AD 683 670

TM-3628/003/00

TECHNICAL MEMORANDUM

(TM Series)

ADVANCED DEVELOPMENT PROTOTYPE

Final Report (Part 1) to the Director,
Advanced Research Projects Agency

for the period
30 July 1968 to 30 January 1969

J. I. Schwartz, Project Director

SYSTEM DEVELOPMENT CORPORATION
2500 COLORADO AVE.
SANTA MONICA
CALIFORNIA
90406

The work reported herein was supported by the Advanced Research Projects Agency of the Department of Defense, and was monitored by the Information Processing Techniques Office under Contract DAHC15-67-C-0149.

Distribution of this document is unlimited.

(page 2 blank)

ABSTRACT

This final report describes work done under Part 1 of the Advanced Development Prototype contract from 30 July 1968 to 30 January 1969. The result of this work is ADEPT--a comprehensive information-processing system implemented at SDC for operation on IBM 360 computers. This report includes an overview of the current status of the system, and a detailed description of the three major components of ADEPT: a time-sharing executive, a data management component (consisting mainly of the Time-Shared Data Management System), and a programmer's package, which includes a JOVIAL compiler editing, debugging, and utility programs, a teletype interpreter (TINT), and an Interactive Programming Support System. Also included in this document are the names of staff members assigned to each of the three major project areas, as well as a listing of the documents produced in each area during this reporting period. Upon request, referenced documents will be made available to appropriate organizations.

CONTENTS

ABSTRACT 1

1. OVERVIEW 5

1.1 Introduction 5
1.2 System Components 5
1.2.1 Time-Sharing Executive 6
1.2.2 Data Management System 6
1.2.3 Programmer's Package 6
1.2.4 Hardware Configuration 6
1.3 Project Status 7
1.4 Organization of Report 8

2. TIME-SHARING EXECUTIVE 9
2.1 Introduction 9
2.1.1 System Architecture 9
2.1.2 Basic Executive (BASEX) 12
2.1.3 Extended Executive (EXEX) 16
2.2 Progress 17
2.2.1 Capabilities 17
2.2.2 Reliability 21
2.2.3 Performance 28
2.3 Staff 30
2.4 Documentation 31

3. DATA MANAGEMENT SYSTEM 37

3.1 Introduction 37
3.2 Progress 38
3.2.1 Define 38
3.2.2 Generate 38
3.2.3 Query 39
3.2.4 Compose 39
3.2.5 Update 40
3.2.6 Maintain 40
3.2.7 Reformat 41
3.2.8 Data Base Oriented Programming Language 41
3.2.9 META5 Adaptation 42
3.3 Staff 43
3.4 Documentation 44

4. PROGRAMMER'S PACKAGE 47

4.1 Introduction 47
4.1.1 JOVIAL Compiler. 47
4.1.2 Debugging Aid. 47
4.1.3 Editing Aid. 47
4.1.4 Utility Program. 48
4.1.5 TINT (Teletype Interpreter). 48
4.1.6 Interactive Programming Support System (IPSS). 48
4.2 Progress 48
4.2.1 JOVIAL Compiler. 49
4.2.2 Debugging Aid. 50
4.2.3 Editing Aid. 50
4.2.4 Utility Program. 50
4.2.5 TINT (Teletype Interpreter). 50
4.2.6 Interactive Programming Support System (IPSS). 51
4.3 Staff. 51
4.4 Documentation. 52

APPENDIX ADVANCED DEVELOPMENT PROTOTYPE SYSTEM BLOCK DIAGRAM. 57

1. OVERVIEW

1.1 INTRODUCTION

Work on the Advanced Development Prototype (ADP) contract was begun in January 1967 for the purpose of demonstrating--in an operational environment--the potential of automatic information-handling made possible by recent advances in computer technology, particularly advances in time-sharing executives and general-purpose data management techniques. The result of this work is a large-scale, multi-purpose system (known as ADEPT), which operates on IBM System 360 computers. The historical background and early development of this system can be traced in the preceding reports in this series.*

The original ADP contract has since been extended, and is currently being performed in two parts: Part 1 (ADP) ended on 30 January 1969; Part 2 (Computer-Aided Command research program) is scheduled to run until 15 September 1970. This report covers work performed under Part 1. (A semiannual technical report on Part 2 will be published in March 1969.)

The entire ADEPT system was operational during this reporting period, and is now being used at four field installations in the Washington, D. C. area, as well as at SDC in Santa Monica. The system was installed at the National Military Command System Support Center in May 1968, at the Air Force Command Post in August 1968, and at two other government agencies in January 1969. These four field sites collectively run ADEPT from 80 to 100 hours per week, providing a total of some 400 hours of time-sharing service monthly to these users. As improved versions of system components become available, they are shipped to the various field sites for installation, which is supervised by SDC personnel.

Information on the ADEPT system was disseminated to the military and governmental communities through a series of ADEPT-50 Symposia. The first symposium, which was held at SDC Santa Monica on 24 and 25 April 1968, was attended by over 200 people. Sponsored by ARPA and SDC, it consisted of a set of briefings on the operation and capabilities of the system, and live demonstrations of major system components. Two more ADEPT-50 Symposia were held at Andrews Air Force Base, Maryland on 10 and 11 July 1968. These two sessions, which were sponsored by SDC, DCA, and OASD Comptroller, were attended by more than 500 people. Briefings and demonstrations on the system were again presented.

1.2 SYSTEM COMPONENTS

The ADEPT system consists of three major components: a time-sharing executive; a data management system adapted from SDC's Time-Shared Data Management System (TDMS); and a programmer's package. Each of these components is discussed in turn below.

---

*TM-3628/000/00, TM-3628/001/00, and TM-3628/002/00.

1.2.1 Time-Sharing Executive

The time-sharing executive of ADEPT consists of two major pieces, one called the Basic Executive (BASEX), the other the Extended Executive (EXEX). BASEX contains elementary functions such as an input/output processor, a scheduler that permits the dynamic adjustment of priorities, an interrupt processor, and a basic sequencer. EXEX contains routines to interpret user commands, file-inventory routines, and various other aids for both programming and nonprogramming users. The Extended Executive is an "open-ended" module that permits expansion when necessary. ADEPT provides for multiple access to the computer through a variety of input/output devices, including typewriter keyboards, small tabular displays, and cathode-ray-tube graphic terminals.

1.2.3 Data Management System

The data management portion of the ADEPT system consists principally of a set of integrated programs designed to handle the most frequently performed data management tasks. Included in this set are programs that allow the user to describe the entries in a data base, load them into the machine, ask questions about them, perform calculations on them, have them presented for his analysis, obtain hard-copy reports, and update and maintain the data base. Several related capabilities were also developed as part of the data management portion of ADEPT; these include a procedure for integrating the data management features of TDMS with the computational capabilities of JOVIAL, and a means for reformatting existing data bases to TDMS format using the META5 language.

1.2.3 Programmer's Package

The third major component of the ADEPT system is designed to provide several powerful tools for programmers with varying skill levels. For the professional programmer, a JOVIAL compiler and a number of service programs (including editing and debugging routines) are provided. For the novice programmer who may occasionally wish to use a computer to solve short, "one-shot" problems, a user-oriented interpreter (TINT) is provided. In addition, an integrated system that provides extensive support to programmers at all levels has been developed as part of this portion of ADEPT. This system--based on SDC's Interactive Programming Support System--will assist machine users in all of the programming processes, ranging from program composition, editing, execution, and testing, through program documentation.

1.2.4 Hardware Configuration

The equipment included in the ADEPT system currently consists of an IBM System 360/50 computer with 262,000 bytes of core memory; three selector channels for transfer of data between the CPU and drum, disc, and tape storage, and a multiplexor channel for transfer of data to and from interactive terminals and other input/output equipment. A block diagram of the current hardware configuration, showing equipment model numbers, speeds, capacities, and device addresses (in hexadecimal), is included as an appendix to this document.

1.3 PROJECT STATUS

During the six-month period ending 30 January 1969, the various ADEPT components continued to operate reliably as an integrated system. At the same time, new capabilities were added to many of the separate components, while program errors and limitations were removed from others. The concurrent shakedown and extension of the system reflected the responses of users at the several ADEPT installations. Their suggestions for improving ADEPT capabilities are being evaluated. A brief overview of the status of the three major ADEPT components is presented in the paragraphs that follow.

Work on the time-sharing executive portion of ADEPT during this reporting period continued to concentrate on adding new capabilities, while at the same time improving performance reliability. Nine more operational releases and experimental pre-releases of the executive were made. The following major capabilities are now provided by the ADEPT executive: a complete file cataloging subsystem, pervasive security controls, an integrated Batch subsystem, dynamic memory allocation, interactive symbolic debugging, support for a variety of interactive devices, and an expanded command library of over 40 interactive (console) commands and numerous program-called executive functions. The "exportability" of ADEPT to new installations was increased by improving techniques for installation and configuration control, system testing and quality control, and continued refinement of system fabrication techniques. Performance of the executive improved during this period, due largely to the use of a new multi-queue scheduling algorithm, a new scheduling command, improvements in LOAD operation, and increased SPAM file size.

TDMS, during this reporting period, evolved into a well defined and controlled system. The functional capabilities of the system are now stabilized, and the system is operating reliably at four military installations. Plans for the addition of further capabilities have been documented. Four new releases of TDMS were made during this reporting period, and one shortly after its close. These releases provided considerable improvements in TDMS components, particularly Generate, Query, and Compose. Work on TDMS is now being concentrated on reducing the processing-time requirements, so that larger data bases can be handled economically. In conjunction with the effort to speed up TDMS operations, some redesign work is being done which will make it possible for TDMS to use the new multi-volume file capability of the ADEPT executive. Finally, continued progress was made in two areas related to the data management portion of the ADEPT system: the data base oriented programming language work, and the adaptation of META5 for data base reformatting.

Work on the programmer's package component of the ADEPT system during this reporting period was devoted largely to shaking down existing programs, removing program "bugs", and documenting program usage. Several components of the programmer's package were used heavily by members of the ADEPT development team in Santa Monica to write and check out system programs. Standard formats for file identification and standard default options were developed and incorporated into

most of the components of the programmer's package. A number of new JOVIAL library procedures were written, many of which are intended to simplify input/output through the Cataloger and SPAM, and to make string manipulation easier. Several new utility programs were written and checked out, and some of the existing utility programs were re-worked. TINT was modified to allow saved programs to be changed using the editing aids provided by IPSS, and work was begun on providing a feature for copying TINT programs to be used later by JOVIAL programmers. IPSS was also improved during this reporting period by "tightening" various parts of the code and by adding several new features, such as better error detection in the editing mode, provision for outputting hard copies of CRT displays, and automatic output of set/use information.

1.4 ORGANIZATION OF REPORT

The organization of this report generally parallels the logical structure of the ADEPT system; each of the three major system components is described in a separate section. Included in each section is a general description of the component, and the progress made in implementing it during the past six-month period. Also included are the names of staff members assigned to the three major project areas, and a list of the documents produced in each area during this reporting period.

2. TIME-SHARING EXECUTIVE

2.1 INTRODUCTION

The ADEPT executive is a general-purpose time-sharing system. The initial system operates on a 360 Model 50 with approximately 260,000 bytes of core memory, 4 million bytes of drum memory, and over 250 million bytes of disc memory, shown graphically in Figure 1. With this machine configuration, ADEPT is designed to provide responsive on-line, interactive service, as well as background service to approximately 10 concurrent user jobs. It handles a wide variety of different, independent application programs, and supports the use of large random-access data files. The design--basically a swapping system--provides for flexibility and expansion of system functions, and growth to more powerful models in the 360 family.

ADEPT functions both as a batch processor (whereby programs are accumulated and fed into the computer for operation one by one) and as an interactive, or on-line, system (in which the user submits his requests directly to the machine in real time simply by typing them on a console). The user first identifies himself through the ADEPT console commands, and specifies his programs and data files.

Viewed as a batch system, ADEPT allows jobs to be submitted to console operators in some standard manner, or submitted from consoles via remote batch commands. In either case, jobs are "stacked" for execution by ADEPT in a first-in/first-out order. The stack is serviced by ADEPT as a background task, subject to the priorities of the installation and the demands of "foreground" interactive users. Viewed as an interactive system, ADEPT allows the user to work with a reactive typewriter, allowing computer-user dialogue in real time. Via ADEPT console commands, the user identifies himself, his programs, and his data files, and selectively controls the sequence and extent of operation of his job in an ad lib manner. A prime advantage of the interactive use of ADEPT is that the system provides for a library of service programs, later extendable, that permit the user to edit data files, compile or assemble programs, debug and eliminate program errors, and generally manage large data bases in a responsive on-line manner.

2.1.1 System Architecture

The architecture of the ADEPT executive is that of the "kernel and the shell." The "kernel," referred to as the Basic Executive (BASEX), handles the major problems of allocating and scheduling hardware resources. It is small enough to be permanently resident in low core memory, permitting rapid response to urgent tasks, e.g., interrupt control, memory allocation, and input/output

traffic The "shell," referred to as the Extended Executive (EXEX), provides the int ace between the user's application program and the "kernel." It contains those non-urgent, large-task extensions of the basic "kernel" processes that are user-oriented rather than hardware-oriented; they may, therefore, be scheduled and swapped.

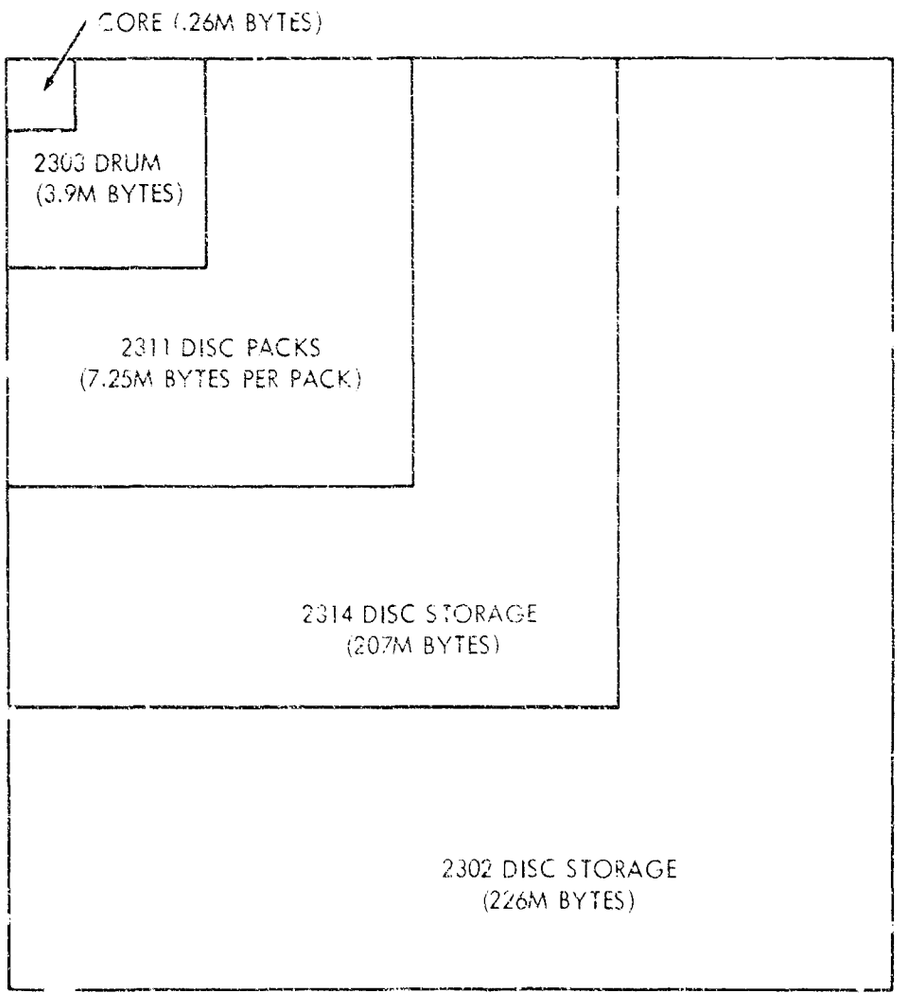


Figure 1. Relative Capacity of Various ADEPT Direct-Access Storage Media Availabl in Less than 0.2 Seconds

The initial system that o erates at SDC utilizes core, 2303 drum, 2311 and 2314 disc packs, and 2303 disc storage. The NMCSSC system successfully utilizes 2314 disc storage in lieu of 2311 or 2302 discs. The architecture of the ADEPT executive is such that it permits any combination of the above types of disc storage in varying amounts.

The version of the ADEPT time-sharing system thus far developed has multiple levels of control beyond the two-level "kernel-shell" structure--i.e., it can be thought of figuratively as an "onion skin." Figure 2 shows these relationships graphically.

Beyond EXEX, "object systems" may exist as subsystems of ADEPT (without modification to EXEX or BASEX), thus further distributing and controlling the resources of object programs that form still another level of the system. The design ideas embodied in ADEPT parallel those of Dijkstra (T.H.E. Multiprogramming System), Corbato (MULTICS), and Lampson (SDS 940 System), but differ in techniques of implementation.

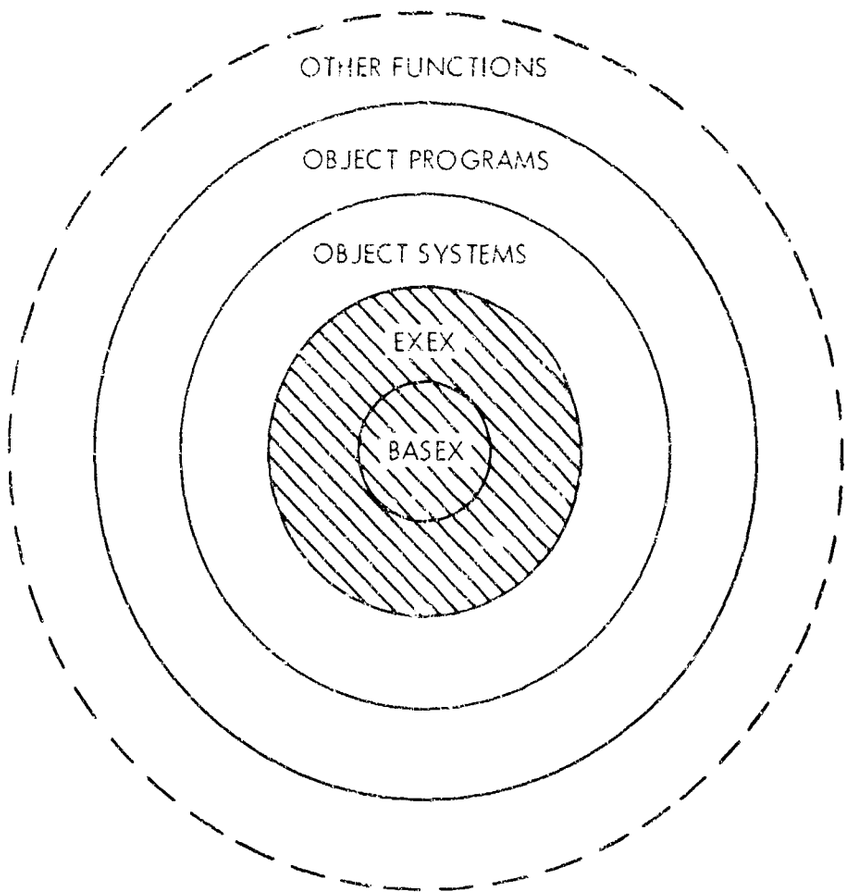


Figure 2. Multiple Levels of Control in ADEPT

The ADEPT executive is designed to operate in the lower quarter of memory, thereby providing three quarters of memory for user programs. With the current hardware configuration, ADEPT preempts the first 65,000 bytes of core memory, the bulk of which is dedicated to BASEX. EXEX then operates in user memory in a fashion similar to user programs. ADEPT is designed to operate itself and user programs as a collection of 4096-byte pages. BASEX is identified as certain pages that are fixed in main storage and that cannot be overlayed or swapped. EXEX and other programs are identified as sets of pages that move dynamically between main storage and swap storage (i.e., drum). It is necessary to maintain considerably more descriptive information about these swappable programs than about BASEX. This descriptive information is carried in a set of system tables that, at any point in time, describe the current state of the system and each program.

ADEPT views the user as a job consisting of some number of programs (up to four for the 360/50H configuration) that were loaded at the user's request. Implicitly, EXEX is considered to be one of these programs. Only one program in the user's set may be active (eligible to run) at a time. When ADEPT scheduling determines that a job may be serviced, the current job in core is saved on swap storage, and the active program of the next job is brought into core from swap storage and executed for a maximum period of time, called a quantum. The process then repeats for other jobs. Figures 3 and 4 schematically depict these relationships.

2.1.2 Basic Executive (BASEX)

Table 1 (below) lists the BASEX components and their general functions as of the seventh and latest executive release. These basic system components form an integrated, non-reentrant, non-relocatable, permanently resident, core memory package 16 pages long. They are invoked by hardware interrupts in response to service requests by terminal users and their programs. Note the division of input/output control into cataloged (SPAM and IOS), terminal (TWRI), and drum (BXEC) activities to permit local optimization for improved system performance.

Table 1. Basic Executive Components

Component	Function
ALLOC	Drum and core memory allocation
BXBUG	Debugger for executive programs
BXEC	Basic sequence and swap control
BECSVC	SVC handlers for WAIT, TIME, DEVICE, STOP and DISMISS calls
EXEX	Linkage routines for EXEX (BASEX/EXEX interfaces); also services commands DIALOFF, DIALON
INTRUP	First-level interrupt control
IOS	Channel-program level input/output supervisory control
SKED	Scheduler
SPAM	Input/output access methods to cataloged storage
TWRI	Terminal input/output control
System Tables	Resident system data areas for communication table (COMTAB), logged-in user's table (JOB), loaded programs table (PGU), drum and core status tables (DSTAT, CSTAT), and a variety of other tables

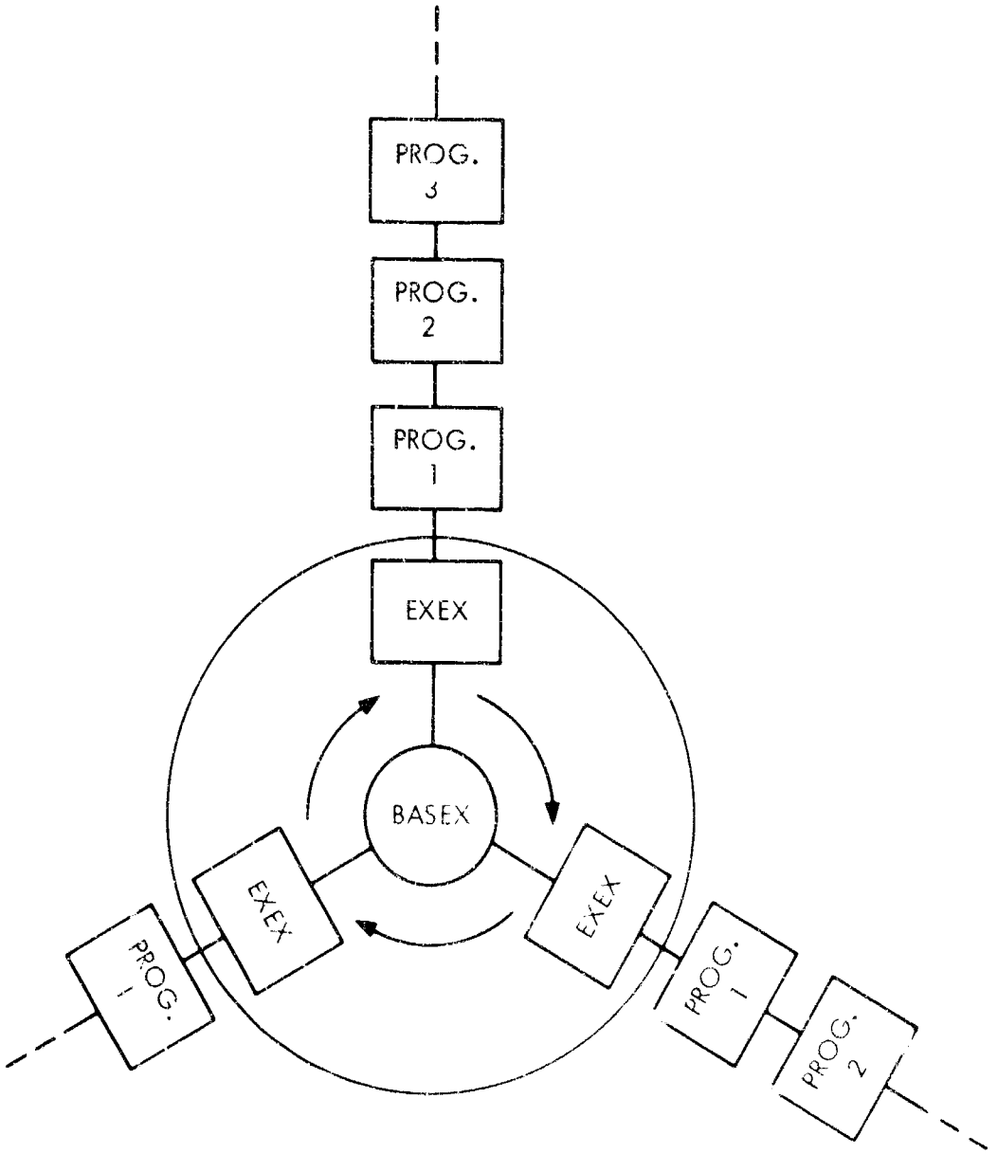


Figure 3. Simple Commutation of User's Programs

This figure illustrates the relationship between user's programs, EXEX, and BASEX. Each spoke represents a user's job, with his EXEX providing the interface between BASEX and the hardware resources. The maximum number of interactive jobs for the IBM 360/50H configuration is ten.

Figure 4. Basic ADEPT Sequence of Operation

This figure shows the basic operating system cycle: idle loop is interrupted by an external interrupt (an activity request); a program is scheduled, swapped into core from the drum, and executed; escape from the execution phase occurs when a quantum termination condition (e.g., time expiration, service or I/O call, error condition) is met; the program is then swapped out and control is returned to the idle loop (if no other programs are eligible to be scheduled).

2.1.3 Extended Executive (EXEX)

Unlike the tight, closed package of integrated BASEX components, EXEX is a loose, open-ended collection of semi-autonomous programs. Table 2 lists this collection for Release 7. EXEX is treated by BASEX as a user program, with certain privileges, and each user is given his own "copy" of the EXEX. It is transparent to the user that EXEX is reentrant and is being shared with other users, except for its data space (the job environment pages are unique for each user). This structure permits flexible modification and orderly expansion in a modular fashion. EXEX is always scheduled like other user programs.

Table 2. Extended Executive Components

Component	Function
BMON	Batch monitor for control of background job execution
CAT	Cataloger for file storage access control; also services FORGET command
DBUG	Debugger for non-executive (user) programs
LOGIN	User authentication and job creation
MBLOAD (or SERVIS)	Library of service commands that are reentrant, interruptible and scheduled: APPEND, CHANGE, CREATE, CYLS, DELETE, DRIVES, INIT, LISTE, LISTU, LOAD, LOADD, LOAD and GO, OVERLAY, REPLACE, RESTORE, RESTORED, SAVE, SEARCH, VARYOFF, VARYON
RUN	Remote batch job submission control servicing commands RUN and CANCEL
XXTOO	Library of small, fast, executive service commands: CPU, BGO, BQUIT, BSTOP, DIAL, DRUMS, GO, LOGOUT, QUIT, RESTART, SKED, SKEDOFF, STATUS, STOP, TIME, USERS
SYSDEF	Defines input/output hardware configuration at time of system initialization
SYSLOG	Defines authorized user/terminal security profiles at time of system initialization

Component	Function
TEST	Initializes system tables at time of system initialization
SYSDATA	Non-resident, shared, system data table for dial messages and other common data, e.g., lists of all logged-in users; other non-resident, job-specific tables also exist, e.g., job environment page, pushdown list data page

2.2 PROGRESS

With the close of this reporting period, development of the experimental ADEPT system under Part I of this contract came to a close. Further experimental work in extending the ADEPT system and related technologies will be carried out under Part 2 of this contract (the Computer-Aided Command research program). Thus this final report on the Part 1 work summarizes ADEPT executive capabilities through Release 7.1.

As in the previous reporting period, the three goals of executive capability, reliability, and performance were pursued in parallel, with primary thrust in the first two areas. Of particular note is the measure of success that has been achieved with the problem of ADEPT "exportability," allowing rapid installation and reliable operation of the system in the field, e.g., at military locations in the Washington, D. C. area.

2.2.1 Capabilities

The capabilities of the ADEPT executive continued to increase in an evolutionary manner through the series of operational releases and experimental pre-releases. The chronological history of these releases up to the present is listed below:

Release	Date
1.0	30 October 1967
2.0	4 December 1967
2.5	15 December 1967
3X	15 January 1968
3.0	12 February 1968
4X	15 April 1968
5.0	20 May 1968
6.3	12 June 1968
7.0	15 July 1968

Release	Date
6.0	29 July 1968
6.1	5 August 1968
6.2	12 August 1968
6.3	9 September 1968
6.4	23 September 1968
6.5	7 October 1968
6.6	Superseded by 7X
7X	31 October 1968
7.0	16 December 1968
7.1	13 January 1969

The current release now possesses the following principal capabilities:

1. A complete file cataloging subsystem. The Cataloger manages 7- and 9- track tape drives, and 2302, 2311, and 2314 discs. Via the Cataloger, a user may maintain his files on private demountable storage volumes (i.e., tapes and disc packs) or on public on-line (POL) storage (i.e., disc). These files may be specified by the owner as Private (for his use only), Semi-Private (for use by a specified community of users), or Public (for general use). Furthermore, access to Public files can be limited by the owner to read-only, write-only, or both read and write; access to Semi-Private files can be limited by the owner to read-only, write-only, or both read and write for each individual in the specified community of users.

   In concert with the Cataloger, system commands CHANGE, CREATE, DELETE, FORGET, INIT, LISTF, and SEARCH permit users to interactively (1) change any of the file descriptors and security control parameters of files they own; (2) create Need-to-Know lists for Semi-Private files they own; (3) purge their files from the file inventory; (4) cancel a mount request for a private tape or disc volume; (5) initialize a scratch 2311 or 2314 disc pack (i.e., clear the volume's catalog and write a volume label) for use with ADEPT; (6) display on their terminals the current inventory of their files, all Public files, or all files on a specified volume; and (7) search the file inventory for the existence and location of a specified file, respectively.

   As the largest single component of the ADEPT executive (65,000 bytes), the Cataloger was written in a new, experimental programming language called MOL-360*, which satisfied the conflicting

---

* MOL-360 (Machine-Oriented Language for the 360) is a "higher-level machine language." It was developed under an ARPA-sponsored SDC research project on metacompilers. (See TM 687/010/00, "Information Processing Techniques Semiannual Technical Summary Report to the Director, ARPA.")

demands for higher-level source language and flexible machine code (i.e., code that provides access to non-standard hardware features). The Cataloger design and checkout was considerably enhanced by the use of MOL-360, while simultaneously showing the validity of MOL compilers for difficult machine-dependent programming. This success has encouraged further experimentation with MOL for coding many system utility tools.

2. Pervasive security controls. Integrated throughout the ADEPT executive are software controls for safeguarding security-sensitive information. The conceptual framework is based upon four "security objects": user, terminal, file, and job. Each of these security objects is formally identified: user by an up to 12-character name; terminal by its hardware address; file by its 8-character name, form (e.g., binary program, card images, etc.), owner-name, and volume number; and job by its internal job-table entry location. Each object is also described by a three-tuplet security property: Classification (e.g., TOP SECRET, SECRET, etc.), Need-to-Know, and Special Category (e.g., SIOP, CRYPTO, etc.). At system initialization time, user and terminal security properties are pre-stored by security officers via the system component SYSLOG. SYSLOG also permits the association of up to 64 passwords with each user. At LOGIN time, a user identifies himself (his unique, up to 12-character name) and enters his private password to validate his identity. The LOGIN component of ADEPT validates the user and dynamically derives the security three-tuplet for the user's job as a complex function of the user and terminal three-tuplets. The job's three-tuplet is subsequently used as "keys" when access is made to ADEPT files. The file's three-tuplet acts as the locks under control of the file subsystem.

File access Need-to-Know is permitted for Private, Semi-Private, and Public use. With the CREATE command, a list of authorized users and the extent of their access authorization (i.e., read-only, write-only, read and write) can be established easily for for Semi-Private files. Newly created files are automatically classified with the job's three-tuplet security property. Through judicious use of the CHANGE command, these properties may be altered by the owner of the file.

Security controls are also involved in the control of classified memory residue. Hardware and software memory protection is extensively used. Software memory protection is achieved by interpretive, legality checking of memory bounds for I/O buffer transfers, legality checking of device addresses for unauthorized hardware access, and other possible user program attempts to seduce the operating system into violating security controls.

3. An integrated Batch subsystem. The RUN command permits on-line users to enqueue jobs into the single ADEPT Batch job stream. Subject to various possible scheduling disciplines, the Batch Monitor component of ADEPT services these jobs in priority order in the ADEPT background. (Interactive terminal-controlled jobs are serviced in the foreground.) Remote job entry is thus permitted.

4. Dynamic memory allocation. As a resource-sharing system, ADEPT provides a wide variety of dynamic memory management capabilities for user and system use. Memory management includes both core and drum memory, allocated in pages of 4096 contiguous bytes. The 2303 drum has an 800-page capacity, whereas the "H" size core memory is viewed as 64 pages. The Allocator component of ADEPT manages a page map for each program. The map reflects the correspondence between drum and core pages, established initially by the ADEPT SERVIS component at LOAD time. User programs may manage their own page maps via a number of Allocator calls. To acquire more drum and core pages than initially loaded, the GETPAGE call may be invoked; to release pages, FREEPAGE call; to permit page sharing, SHARE call; to modify (activate or de-activate) the swappable set of program pages, ACTDEACT call; and to copy the page map, the ESTACOP call. The user program may also load additional pages from disc or tape via the SERVIS calls APPEND and OVERLAY. Via these calls, skilled users can achieve efficient use of time and memory. Most EXEX components use these calls for just such purposes.

   The single most important performance aspect of ADEPT is in the management of the drum memory by the Swapper component. By marking only those pages changed (i.e., written onto) during a program's time-slice, considerable reduction in swap time has been attained. This technique for efficiently managing memory is further described below (see Section 2.2.3).

5. Interactive symbolic debugging. ADEPT JOVIAL programmers can debug their programs on-line using symbolic item names and statement labels as address parameters for the Display, Set, Breakpoint, and Goto DBUG commands. This has been achieved by allowing a compiler-produced dictionary to be loaded with the program as the result of the SERVIS component LOADD command. DBUG has been modified to use this dictionary for all symbolic address references. DBUG can also be used for debugging in machine language by use of absolute memory addresses and programmer-defined display formats.

6. Interactive devices. The prototype system at SDC has been expanded to permit the use of a variety of interactive terminals. It is possible to support Model 33/35 Teletypes, both local and remote (hardwired or dialup), IBM 2741 and 1052 typewriters, IBM 2260 and CCI cathode-ray tube devices. The amount of memory, however, limits the number and variety of such terminals that can be accommodated by a given system configuration. With all these devices, users may correct typographical errors by line- or character-level cancellation. In addition to these devices, ADEPT supports the large IBM 2250 graphic display and the IBM 1053 hardcopy printer for the 2260 CRT.

7. Command Library. Over 40 interactive commands are available to users. Although the three commands of LOGIN, LOAD (and) GO, and LOGOUT are all that is necessary to use ADEPT, the expanded command vocabulary provides more knowledgeable users greater power and flexibility in their use of the system. In addition to these console commands, a variety of service calls (SVC's) are available to programs. Table 3 summarizes both console and program calls.

2.2.2 Reliability

After achieving executive capabilities beyond those minimums needed for a meaningful user environment (i.e., Release 3), emphasis was shifted toward attaining higher system reliability. This goal was translated into the following activities:

1. System fabrication. Better control was gained over the production of new releases, thus minimizing errors of omission and commission. The Load and Initialization Package (LIP) continued to be the principal tool for system fabrication. Work on LIP focused on cleaning up old features to make them work better or simpler, and adding new capabilities. In the former area, the structure of the master system deck was simplified by the elimination of extra, unnecessary control cards. In the latter area, LIP clears core (a confusing set of operator actions was previously required); it also clears the protection keys prior to system loading. LIP permits the dumping of disc and core memory, and the listing of all system components.

2. Installation and configuration control. Work has gone into greater parametric control of system decks for producing systems for different hardware configurations (e.g., NMCSSC has no 2302, and Air Force Command Post has no 2741 terminals).

Table 3. Summary of ADEPT Executive Functions

Function	Type of Component	Called (External/Internal)	CVC Number* Dec.	Hex.	Description
ACTDEACT	B	I	17	11	Activate or deactivate program pages.
APPEND	X	E,I	35	13	Load a new program into the next available core page immediately following the initially loaded program.
B (break)	X,C	E	39/40	27/28	(A debugging command.) Insert or remove a breakpoint in user's program.
BGO	X,C	E	--	--	This command can be issued from the operator's terminal only. Initially start the batch monitor. This command resets the batch monitor PCW to the program's initial location. It should not be used after a BSTOP command.
BQUIT	X,C	E	--	--	This command can be issued from the operator's terminal only. Quit the present batch job, but continue processing.
BSTOP	X,C	E	--	--	This command can be issued from the operator's terminal only. Stop the Batch Monitor according to the conditions specified by the command programmer.
CANCEL	X,C	E	48	3	Cancel a batch job that has been previously requested by the RUN command.
CATALOG	X	I	[illegible]	[illegible]	Call the EXEX Cataloger to open, close, change, or delete tape and disc files.
CHANGE	X	E	34	[illegible]	Change parts of the description of a cataloged file.**
CHANGE	X	I	[illegible]	26	One of the CATALOG functions. Change a file.
CLOSE	X	I	38	26	One of the CATALOG functions. Close a file.
CPU	X,C	E,I	19	13	Provide information on the amount of central processor time used by the named program. The swap time is not included.
CREATE	X	E	34	22	Attach a need-to-know list to the file, fcnf. The file, fcn, described by the parameters preceding the comma, is a list of legal *uscrids* with the access allowed for each user. The file, fcn, consists of a set of cards (form = F-). Each card contains a *uscrid* followed by one or a combination of the following letters: A, T, R, W.
CYLS	X	E	34	22	List the total number of available cylinders on the specified device.*
CYLS	X	I	38	26	(One of the CATALOG functions.) List of available cylinders.
D (dump)	X,C	E	39	27	(A debugging command.) Display contents of specified portions of user's program.
DBUG	X,C	E	39	27	Request use of the EXEX debugging commands.
DELETE	X	E	34	22	Delete a file.**
DELETE	X	I	38	26	(One of the CATALOG functions.) Delete a file.

B = BASEX, X = EXEX, C = Command Library, E = externally called via console command, I = internally called via CVC.

* Functions in the Command Library that are permanently resident BASEX or EXEX components are not assigned CVC numbers.

** Same function as below except this one is externally called.

Table 3. Summary of ADEPT Executive Functions (Con't.)

Function	Type of Component	Called External/ Internal	SVC Number* Dec.	Hex.	Description
DEVICE	B	I	22	16	Provide the current time, date, computer model, size of core, software release version, type of terminal, whether or not device is enabled, and the on-off status of the various devices attached to the system.
DIAL	X,C	E	--	--	Send a message from one console to another.
DIALOFF	X,C	E	--	--	Suppress DIAL messages sent from other consoles. DIAL messages are normally allowed.
DIALON	X,C	E	--	--	Accept DIAL messages sent from other consoles. Clear the "not accepting message" indicator which is set by the DIALOFF command.
DISMISS	B	I	31	1F	End the quantum. Immediately reschedule the program for the next quantum.
DRIVES	X	E	34	22	Ascertain the number of available tape or private disc drives available**
DRIVES	X	I	35	23	(One of the CATALOG functions.) Provide information on available drives.
DRUMS	X,C	E, I	--	--	Provide information on available drum space.
EXCP	B	I	17	11	Execute a channel program.
FORGET	X	E	--	--	Cancel a request to the Cataloger for a tape or disc volume. This command causes a no-wait read on the terminal.
FREEPAGE	B	I	15	0F	(One of the Allocator functions.) Release page(s).
T (go to)	X,C	E	39	27	(A debugging command.) Specify a point in user's program at which to begin or resume execution.
GETPAGE	B	I	14	0E	(One of the Allocator functions.) Assign new page(s).
GO	X,C	E	45	2D	Initiate program execution. Stop executing (this) calling program and go to execute the named program.
INIT	X	E	32	20	(This command can be issued from the operator's terminal only). Clear the Cataloger tables from the specified disc, and label it with the volume number specified.
LISTF	X	E	36	24	Obtain a list of the specified files on the specified volumes.**
LISTF	X	I	38	26	(One of the CATALOG functions.) Obtain a list of the specified files.
LISTU	X	E	34	22	List all of the users currently logged into the system and the terminal address at which they are logged in.
LOAD	X	E,I	34	22	Load a program at location 10,000 (hexadecimal) and following.
LOADD	X	E,I	34	22	Load a program and a dictionary from the same unit.
LOADGO	X	E,I	44	2C	Load a program and initiate program execution (i.e., LOAD and GO feature). Command names are searched first. The user should avoid program names that clash with system commands.
LOGIN	X	E,I	47	2F	Log into the time-sharing system. This function is internally executable by the Batch Monitor.
LOGOUT	X,C	E,I	53	35	Exit from the system. Remove all active programs in job. Total quit
OPEN	X	I	38	26	(One of the CATALOG functions.) Open a file.
OVERLAY	X	E,I	36	24	Load a new program at the lowest core page assigned to an initially loaded program
PRINT	B	I	08	08	Print a line of text on the user's IBM 2741, Model 33 or 35 Teletype, or IBM 1050
QUIT	X,C	E,I	40	28	Remove a program from the job (and system). Partial logout.

B = BASEX, X = EXEX, C = Command Library, E = Externally called via Console command, I = Internally called via SVC.

* Functions in the Command Library that are permanently resident BASEX or EXEX components are not assigned SVC numbers.

** Same function as below except this one is externally called.

Table 3. Summary of ADEPT Executive Functions (Con't.)

Function	Type of Component	Called (External/ Internal)	SVC Number Dec.	Hex.	Description
READ	B	I	09	09	Read a line of text from the user's IBM 2741, Model 33 or 35 Teletype, or IBM 2260.
REPLACE	X	E	34	22	(This command can be issued from the operator's terminal only.) Replace on-line terminal with an off-line terminal. Turn the first unit (terminal) off-line and replace it with the second unit.
					Change security and category data of a unit.
					Move a job from one unit to another.
RESTART	X,C	E	--	--	(This command can be issued from the operator's terminal only.) Cause a "break key" operation for a specified terminal. Useful for awakening terminals hung up in erroneous I/O pause.
RESTORE	X	E, I	44	2C	Restore a program that was previously saved.
RESTORED	X	E, I	44	2C	Restore a program that was previously saved and its associated dictionary, if any.
RETURN	X,C	I	46	2E	Return to the original calling program.
RUN	X,C	E	48	30	Enqueue program for batch execution.
S (set)	X,C	E	39	27	(A debugging command.) Alter the contents of specified portions of user's programs.
SAVE	X	E,I	43	2B	Save an initially loaded program on a specified device.
SEARCH	X	E	34	22	Locate file in system cataloger.
					Determine on which disc the file, *fdn*, is stored. If the volume is not specified, all POL volumes are searched. **
SEARCH	X	I	38	26	(One of the CATALOG functions.) Locate file in system cataloger.
SERVIS	X	E,I	34	22	Call the FXEX SERVIS routine to locate programs, save programs, change file descriptions, list files, list users, change equipment status, and interrogate status of equipment. See specific functions such as LOAD, SAVE, RESTORE, CHANGE, etc.
SHARE	B	I	16	10	(One of the Allocator functions.) Share pages of (this) calling program by assigning some or all of its pages to other programs.
SKED	X	E			(This command can be issued from the operator's terminal only.) Force top priority scheduling to one specified job.
SKEDOFF	X	E			(This command can be issued from the operator's terminal only.) Return privilege job to normal scheduling state.
SPAM	B	I	11	0B	Call SPAM to read, write, modify, delete, position, or search for records on a tape or disc file.
STATUS	X,C	E	--	--	Provide information on the status of named program (*program-name*) or job which has the specified terminal (*terminal*) associated with it.
STOP	B,C	E	--	--	If there is no *program-name* parameter, stop current active program, otherwise, stop named program.
STOP	B	I	00	00	Stop current active program.
SWAP	X,C	I	54	36	This call is used for forced swapping of program pages that contain machine instructions which cannot be handled by the page marking routine. If a page is specifically marked as as result of this call, it will be swapped from core to the drums at the end of user's quantum.
TIME	B,C	E,I	19	13	Provide the current time and date.
USERS	X,C	E	--	--	Provide information on the current number o. ADEPT users.
VARY	X	E	34	22	(This command can be issued from the or ator's terminal only.) Varies non-terminal devices on- off-line.
WAIT	B	I	02	02	Wait for time. This call specifies a mi mum interval before the calling program should be rescheduled.

B = BASEX, X = FXEX, C = Command Library, E = externally called via console command, I = internally called via SVC.
* Functions in the Command Library that are permanently resident BASEX or FXEX components are not assigned SVC numbers.
** Same function as below except this one is externally called.

The system will accept 2303, 2311, and 2314 discs, 7- and 9-track tapes, and a wide variety of terminals.

An ADEPT executive initilization function known as SYSDEF provides a capability for specifying the input/output hardware configuration of the system by means of control cards read in at the time of system initilization. The control cards are placed at the head of the run deck from which they are read by LIP and communicated to SYSDEF via the TEST component of EXEX. SYSDEF scans the cards and initilizes the system tables and items for components TWRI, IOS, CAT, BXEC, and INTRUP. These data define the quantity, type, and location (i.e., device address) of drums, terminals, tapes, discs, unit record equipment, and other miscellaneous devices. SYSDEF also designates the address of the operator's terminal, and which devices shall be permanent on-line (POL) disc storage for the Cataloger.

Considerable effort was expended to ease the operator's task in initiating the system from a cold start. This process now takes a maximum of five minutes, if every option is exercised. Communication with the operator via the 1052 typewriter was simplified. He has a number of options, which include varying devices on/off-line, initiating hardware status checks, clearing and testing the drums, initiating the security SYSLOG procedures, and setting the date and time of day.

3. System testing and quality control Since computer system development is still an art, reliability can be achieved only through empirical analysis and operation. System errors generally fall into one of four categories: design, coding, fabrication, or hardware errors. Effective measures have been instituted in each of these areas to promote effective system checkout and high system operating reliability.

Design errors are the most difficult to rectify, particularly context-dependent design errors which appear intermittently during system operation. Such errors are difficult to isolate inasmuch as the error context is nearly impossible to replicate a posteriori. Techniques such as recording and trace histories are employed in checkout, but these techniques have limited utility during operational use due to their high overhead costs in time and storage resources. The decision to incorporate a debugging capability (BXBUG) into the resident (BASEX) executive, however, has paid handsome dividends. New releases can be intimately probed on-line for cause and effect, whenever system errors arise; often the trouble can be fixed immediately, without requiring a system restart. BXBUG is our first line of defense for diagnosis and repair of context-dependent software errors.

Our second line of defense is the Discrepancy Report. This procedural tool imposes a formal error reporting structure on all system users. The cumulative data thus generated narrows the search-space for the error's context. This procedure was first installed for the Release 7 executive and has helped isolate over fifty errors, most of which were quickly resolved and repaired.

Coding and fabrication errors are much more easily detected than design errors. They are simple to detect in the listings during pre-release checkout. Their causes include clerical mistakes, incorrect use of fabrication tools (e.g., assembler, loaders, LIF, etc.), and poor fabrication control. The development of a stable set of benchmark and quality control tests has aided in this task. Also, establishment of procedural controls for the handling and maintenance of symbolic assembly modules and run decks has lowered the incidence of fabrication errors and errors induced by attempted fixes of other errors. It has also provided in-depth backup in the event of damage or loss of component files, and has provided a formal mechanism for generating current system listings.

Hardware errors are now less catastrophic than in the last reporting period. The principal reason has been the incorporation of retry capability into the input/output components of the ADEPT executive. Additional software to better sense and handle recoverable errors (including both hardware and user abuses) also contributes to the improvement. Now, rather than causing a system crash, only the user-job affected is aborted, while other users continue unperturbed and often unaware of the difficulties. Finally, better pre-conditioning of the system at initialization time often avoids trouble later on with marginal components. This is particularly true for marginal drum tracks which are communicated to the Allocator so it can avoid their allocation.

Finally, operation of the system in the face of localized peripheral equipment difficulties is now possible by reconfiguration of the system at the time of initialization with SYSDEF, and dynamically during system operation with the VARY, REPLACE, and RESTART commands.

4. Operational reliability data. Table 4 shows ADEPT operational statistics for the Santa Monica 360/50 over a seven-month period. One major trend is apparent: as the ADEPT executive grows more complex with successive versions, measures of "run time" and "time till failure" tend to get lower and lower. Correspondingly, each new version requires more extensive and longer checkout for appropriate program quality control. For example, the first entry, "Mean Run Time," shows a steady decline in system performance from

Table 4. Experimental Santa Monica ADEPT-50 Operational Statistics
July 1968 through January 1969

Variable	Jul.	Aug.	Sept.	Oct.	Nov.	Dec.	Jan.
Mean Run Time (min)	86	127	71	71	48	65	76
Mean Time Till Failure	68	100	63	60	37	43	69
Median Time Till Failure	62	74	40	27	22	21	40
Median, End-of-Day Run	125	166	75	86	60	87	123
% Hardware Terminations	31	30	23	27	39	15	14
% Software Terminations	31	34	56	42	39	54	31
Mean Recovery Time (min)	6	6	5	5	7	5	4.6
N(Sample of Computer Runs)	58	47	78	68	76	64	44
ADEPT Release	4	6.1	6.2	6.4	6.4	7.0	7.0
	5	6.2	6.3	6.5	7.0		7.1
			6.4	7.0			

August through November. In December, when version 7.0 was checked out for a relatively extended period, there was a substantial improvement from a low of 48 minutes in November to 65 minutes in December. This improvement continued through January with the run-time variables showing increases of 15 to 100 percent over their values in December. ADEPT performance for January was significantly improved and is about the same as it was for September. The median time for the end-of-day run remains two to three times higher than the median time till failure, thus indicating the continuing existence of a daily system shakedown effort. Note also that the percent of software failures has dropped significantly.

These operational data underline the value of stringent and comprehensive quality control before a model is released, together with a somewhat more relaxed pace for incorporating major new features than has been the case in the past.

The magnitudes of the numbers in Table 4 are low, but not surprising, since these data were generated with the experimental SDC Santa Monica hardware-software ADEPT configuration that is used for system development. This system is constantly undergoing changes, as shown by the variety of ADEPT releases for which data were collected. Typically, these releases undergo months of operational use at Santa Monica before being shipped to field installations. Initial reports from ADEPT field installations corroborate the expected high reliability of the executive. Although recording similar to that performed at SDC is not performed by field installations, typical runs of two to four hours are the rule, limited usually by the facility's schedule, and not the system's performance.

As of this writing, ADEPT Release 7.0 is operating at four field installations in the Washington, D. C. area. The first field installation of ADEPT, at the National Military Command System Support Center (NMCSSC), took place in late May 1968, with formal operation the first week in June, as scheduled. The second field installation, at the Air Force Command Post (AFCP) took place in early August 1968. The third and fourth systems were brought up in January 1969 at two other government agencies. These four sites each run ADEPT 20 to 25 hours weekly, providing some 400 console hours of time-sharing service monthly (total).

Other "tests by fire" for ADEPT were achieved at three successful ADEPT-50 Symposia, at which live system demonstrations were given to hundreds of interested military and civilian personnel. The symposia were held at SDC, Santa Monica on April 24 and 25, and at Andrews Air Force Base, Maryland, on July 10 and 11.

2.2.3 Performance

Improving the system's performance is a continuing objective. Efforts are being directed toward balancing the equation of fast response time and high throughput via improved scheduling. However, improving performance also requires that we pay attention to details of user facilities--smoothness of man-machine dialog. Finally, performance improvement implies continual reflection on internal system design in an ongoing effort to lower system overhead. Measurement and recording are the latest tools employed in these efforts.

In contrast to Release 2, Release 7.0 provided an approximate five-fold improvement in throughput at a nominal cost in response time. The specific system changes responsible for this improved performance include:

1. Dynamic page marking. Whenever a user program is swapped into core, its pages are set in a read-only condition. As the program executes, it periodically attempts to store date (write) in its write-protected pages. The resulting interrupt is fielded by

the system. After satisfying itself that the store is legal for the program, the executive marks the target page as "written," turns off the write protect for that page, and resumes the program's execution. The situation repeats for each additional page written. At the completion of the program's time slice, the Swapper has a complete map of all the program's pages that were changed. Only the changed pages are swapped out of core. Preliminary measurement of this scheme shows that about 20 percent of the pages are changed, and hence for every five pages swapped in, only one need be swapped out, for a total swap of six pages, rather than the full swap of ten pages (five in, five out). The scheme thus makes the drum appear to be 40 percent faster.

2. Improved scheduling. The current time slice is now a full two seconds. This raises response time to about five seconds, but it improves system throughput by lowering the swap frequency (and hence swap overhead). In addition, programs are now rescheduled after an I/O request, if there is sufficient time remaining in the time-slice. This technique has improved 2260 service dramatically. The scheduling algorithm now employs an improved multi-queue discipline.

The Release 7.0 scheduling algorithm was modified to handle two levels of scheduling. Jobs that are in a "terminal I/O complete" state get first preference in the schedule. Jobs in the second level or background queue are run if there are no level-one jobs to run. A job is placed in level two when the two-second quantum time terminates its operation two consecutive times. Compute and I/O bound programs are treated the same. A level-two job--when allowed to run--is given a quantum interval equal to the basic quantum time multiplied by the scheduling level (i.e., 2 sec x 2 = 4 sec). However, a level-two background job may be preempted after two seconds for terminal I/O. Any operation a level-two job makes which terminates its quantum prematurely will return it to a level-one status. A Batch Monitor job never sinks below a level-two state. This new algorithm has resulted in a noticeable improvement in Batch Monitor job throughout and terminal response time for the full complement of 10 users.

A new command, SKED, which is limited to the operator's terminal, has been implemented. It has the effect of forcing top priority for a job (the job stays at level one all the time). Only one such job may run in this privileged scheduling state at a time. The command form is:

/SKED *terminal:number*
/SKEDOFF

SKEDOFF removes the job from privileged scheduling.

3. Buffered output. Interactive terminal output is now buffered by the system, thereby permitting some overlapped I/O and program execution.

4. Priority SVC's. Priority processing of selected Allocator calls has yielded a four-fold improvement in the speed of the LOAD command, which makes heavy use of these calls. LOAD now loads programs, eight pages at a time. With these changes it takes about 30 seconds to load a large program (30 pages) with heavy system usage. In addition, user programs are permitted a drum memory of 128 pages, with 256 pages total, per job. A job may consist of up to four programs (including EXEX).

5. SPAM improvements. Recent improvements have lowered the overhead for "position type" I/O calls. With these improvements, SPAM (the ADEPT Sequential Partitioned Access Method) can calculate and position disc heads directly to the proper file record. Previously, a sequential search through all previous records on a track was required. SPAM has also been modified to permit files containing more than 256 logical cylinders and to permit I/O transfers of up to 65,000-byte records.

2.3 STAFF

C. Weissman, Staff Head
C. E. Fox, Technical Assistant
S. M. Aranda, Project Leader
R. R. Linde, Project Leader
H. Sackman, Project Leader

P. S. Baker
Martha Bleier
R. A. Brewer (part time)
R. A. Fisher
B. D. Gold
A. Karush
J. E. Kreznar
Pat Kribs
R. E. Martin
F. M. Stepczyk
S. G. Swerdlow
A. Tschekaloff

2.4 DOCUMENTATION

This section lists those documents describing the executive portion of the ADEPT system that were produced during this reporting period. Some of the documents listed below were released in SDC's "Note" series. These are internal working papers only, and have not been cleared for open publication.

Baker, P. and Tschekaloff, A. Specifications for new ADEPT file structures. SDC document N-23741/071/00. 6 January 1969. 15 pp.

Describes a capability for handling multi-volume file structures.

Bleier, M. B. Reentrant public housefiles. SDC document N-(L)-23741/131/00. 15 August 1968. 8 pp.

Describes the initial design criteria and requirements for reentrant public housefiles.

Kennedy, P. R. User's guide for the ADEPT-50 time-sharing system, the programmer's package, and miscellaneous utility programs--Introduction to the series and table of contents. SDC documents N-23759/000/07 through 10. 30 July 1968 through 30 January 1969. 4 pp. each.

Introduces a document series that provides information on the use of ADEPT system components; includes volume numbers, titles, and current issue information.

Aranda, S. M. and Baker, P. S. ADEPT cataloger user's guide. SDC document N-23759/002/03 and /03A. 8 October 1968. 50 pp.

Describes procedures to be followed in using the ADEPT Cataloger; includes a discussion of Cataloger call procedures, request table formats, and special terms used in connection with the Cataloger.

Tschekaloff, A. ADEPT SPAM user's guide. SDC document N-23759/004/03. 25 September 1968. 28 pp.

Provides instruction for ADEPT users on how to use SPAM for reading, writing, altering, positioning, and searching for records in disc- and tape-based file structures. Includes a discussion of procedures for making SPAM calls, and request table formats.

Martin, R. ADEPT BXBUG user's guide. SDC document N-23759/005/00. 21 November 1968. 10 pp.

Describes how to use BXBUG, a BASEX component that provides a debugging facility and input/output routine for the IBM 1052 console typewriter (i.e., the typewriter used by the computer operator).

Bleier, M. B. ADEPT SERVIS user's guide. SDC document N-23759/021/04. 10 October 1968. 57 pp.

Provides instruction for ADEPT users on how to use SERVIS, that portion of the ADEPT-50 executive that enables users to load programs, save programs, change file descriptions, list files, list users, change equipment status, and interrogate the status of equipment. The on-line commands and internal calls to SERVIS are described.

Bleier, M. B. ADEPT LOGIN user's guide. SDC document N-23759/023/01. 25 November 1968. 7 pp.

Instructs the prospective ADEPT Time-Sharing System user in how to use the LOGIN command.

Kennedy, P. R. ADEPT user's reference card--release 7. SDC document N-23759/061/00 and /01. 14 November 1968. 2 pp.

Provides the experienced ADEPT user with a quick reminder of system command formats and other pertinent information.

Kribs, P. SYSDEF user's guide. SDC document N-23759/703/00. 7 November 1968. 6 pp.

Describes SYSDEF, which is an ADEPT initialization function that provides a capability for specifying the input/output configuration by means of control cards at system load time.

Martin, R. ADEPT-50 release 7.0 operator's guide. SDC document N-23759/704/00. 12 December 1968. 13 pp.

Provides the computer operator and system programmer with information for running and loading ADEPT 7.0.

Kennedy, P. R. ADEPT-50 time-sharing system bulletin: Introduction to the series and table of contents. SDC document N-23853/000/03. November 1968. 2 pp.

Base volume of the N-23853 series. The bulletins in this series announce pertinent information concerning the ADEPT-50 time-sharing executive. This volume contains a table of contents for the entire series.

Bleier, M. B. ADEPT-50 time-sharing system bulletin: Number 7--New SERVIS command formats for Release 7.0. SDC document N-23853/007/00. 10 October 1968 4 pp.

Describes new SERVIS command formats.

Kennedy, P. R. ADEPT-50 time-sharing system bulletin: Number 8--Summary of ADEPT functions through Release 7.0. SDC document N-23853/008/00 and /01. 15 November 1968. 30 pp.

Summarizes all the ADEPT executive functions through Release 7.0.

Kennedy, P.R. ADEPT-50 time-sharing system user's guide: Contents. SDC document TM-3881/001/00. 26 December 1968. 12 pp.

Base volume for the TM-3881 series, which describes the use of the ADEPT-50 Time-Sharing System, including the programmer's package.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: Summary of ADEPT-50 executive functions. SDC document TM-3881/007/00. 26 December 1968. 31 pp.

Summarizes the executive functions available in the ADEPT-50 Time-Sharing System.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--APPEND. SDC document TM-3881/016/00. 8 January 1969. 11 pp.

Describes the use of executive function APPEND.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--CANCEL. SDC document TM-3881/032/00. 23 January 1969. 3 pp.

Describes the use of executive function CANCEL.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--CATALOG. SDC document TM-3881/034/00. 23 January 1969. 5 pp.

Describes the use of executive function CATALOG.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--CHANGE. SDC document TM-3881/038/00. 2 January 1969. 7 pp.

Describes the use of executive function CHANGE.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--CLOSE. SDC document TM-3881/040/00. 23 January 1969. 4 pp.

Describes the use of executive function CLOSE.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--CPU. SDC document TM-3881/042/00. 27 January 1969. 3 pp.

Describes the use of executive function CPU.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--CREATE. SDC document TM-3881/044/00. 2 January 1969. 5 pp.

Describes the use of executive function CREATE.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--CYLS. SDC document TM-3881/046/00. 2 January 1969. 7 pp.

Describes the use of executive function CYLS.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--DELETE. SDC document TM-3881/052/00. 16 January 1969. 9 pp.

Describes the use of executive function DELETE.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--DIAL. SDC document TM-3881/050/00. 27 January 1969. 3 pp.

Describes the use of executive function DIAL.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--DIALOFF. SDC document TM-3881/060/00. 27 January 1969. 3 pp.

Describes the use of executive function DIALOFF.

Kennedy, P. R. ADEPT-50 time sharing system user's guide: ADEPT-50 executive function--DIALON. SDC document TM-3881/062/00. 28 January 1969. 3 pp.

Describes the use of executive function DIALON.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--DRIVES. SDC document TM-3881/066/00. 16 January 1969. 5 pp.

Describes the use of executive function DRIVES.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--DRUMS. SDC document TM-3881/068/00. 29 January 1969. 3 pp.

Describes the use of executive function DRUMS.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--FORGET. SDC document TM-3881/076/00. 30 January 1969. 3 pp.

Describes the use of executive function FORGET.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--GO. SDC document TM-3881/084/00. 30 January 1969. 3 pp.

Describes the use of executive function GO.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--INIT (Operator's Console Command). SDC document TM-3881/086/00. 16 January 1969. 6 pp.

Describes the use of executive function INIT.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--LISTF. SDC document TM-3881/088/00. 22 January 1969. 8 pp.

Describes the use of executive function LISTF.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--LISTU. SDC document TM-3881/090/00. 17 January 1969. 3 pp.

Describes the use of executive function LISTU.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--LOAD. SDC document TM-3881/092/00. 17 January 1969. 13 pp.

Describes the use of executive function LOAD.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--LOADD. SDC document TM-3881/094/00. 20 January 1969. 4 pp.

Describes the use of the executive function LOADD.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--LOADGO. SDC document TM-3881/096/00. 20 January 1969. 4 pp.

Describes the use of the executive function LOADGO.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--LOGIN. SDC document TM-3881/098/00. 30 January 1969. 9 pp.

Describes the use of the executive function LOGIN.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--OPEN. SDC document TM-3881/104/00. 24 January 1969. 26 pp.

Describes the use of the executive function OPEN.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--OVERLAY. SDC document TM-3881/106/00. 20 January 1969. 13 pp.

Describes the use of the executive function OVERLAY.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--REPLACE (Operator's Console Command). SDC document TM-3881/114/00. 22 January 1969. 7 pp.

Describes the use of the executive function REPLACE.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--RESTORE. SDC document TM-3881/118/00. 21 January 1969. 4 pp.

Describes the use of the executive function RESTORE.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--RESTORED. SDC document TM-3881/120/00. 22 January 1969. 4 pp.

Describes the use of the executive function RESTORED.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--SAVE. SDC document TM-3881/128/00. 23 January 1969. 10 pp.

Describes the use of the executive function SAVE.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--SEARCH. SDC document TM-3881/130/00. 22 January 1969. 9 pp.

Describes the use of the executive function SEARCH.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--SERVIS. SDC document TM-3881/134/00. 8 January 1969. 4 pp.

Describes the use of the executive function SERVIS.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--VARY (Operator's Console Command). SDC document TM-3881/154/00.

Describes the use of the executive function VARY.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--SERVIS. SDC document TM-3881/134/00. 8 January 1969. 4 pp.

Describes the use of the executive function SERVIS.

Kennedy, P. R. ADEPT-50 time-sharing system user's guide: ADEPT-50 executive function--VARY (Operator's Console Command). SDC document TM-3881/154/00. 22 January 1969. 7 pp.

Describes the use of the executive function VARY.

Kennedy, P. R. ADEPT-50 time sharing system user's guide: TINT1.1 user's guide. SDC document TM-3881/502/00. 1 August 1968. 179 pp.

A self-instructional guide to TINT1.1, an interactive programming system operating under control of the ADEPT-50 executive.

3. DATA MANAGEMENT SYSTEM

3.1 INTRODUCTION

The data management component of the ADEPT system consists principally of an adaptation of the Time-Shared Data Management System (TDMS), combined with some additional tools in the areas of a data base oriented programming language, and the specialization of META5 to symbolic file conversions for TDMS.

During this reporting period, TDMS evolved into a well defined and controlled system. The functional capabilities of the system have been documented, along with new features scheduled for release in subsequent versions of TDMS. The latest release, Version 5, was sent to the SDC Falls Church office as scheduled, shortly after the close of this reporting period. This release was installed at the various military facilities using TDMS. Only one feature planned for inclusion in Version 5 was delayed; this feature is expected to be shipped as part of an interim release within a few weeks.

Each succeeding version of TDMS has proved to be more reliable than the preceding one, and the system is now in use at four military installations in the Washington, D. C. area. The versions released during this reporting period and their release dates are:

Version	Date
3	6 September 1968
3.1	16 October 1968
3.2	1 November 1968
4	27 November 1968
5	6 February 1969

In addition to the many improvements made to the system itself, procedures were established to handle discrepancy reports from users, and assure that discrepancies are answered and errors corrected. Current reactions to TDMS by a small user population indicate that it is now operating with good reliability and that it performs a variety of desired functions. The prime limitation at present seems to be the speed of operation.

Work on TDMS will continue under Part 2 of this contract (the Computer-Aided Command research program). More of the emphasis on TDMS will be placed on reducing processing-time requirements, particularly for the Generate operation. Improving the speed of Generate is expected to provide the capability to handle larger data bases in less time than is now possible. Also, the form of data bases is now being changed so that all records are of a single, fixed length, thus making possible the use of the new multivolume file capability of the ADEPT executive.

3.2 PROGRESS

TDMS consists of an integrated set of operations, each one of which is designed to do some part of the total data management task facing a user. The operations delivered with the ADEPT system are Define, Generate, Query, Compose, Update, Maintain, and Reformat. In addition, several utility programs useful in testing and analysis of TDMS data bases have been created.

The sections that follow discuss the major TDMS operations, the progress made in implementing them, current capabilities and limitations of the system, and progress made in implementing the other (non-TDMS) data management tools.

3.2.1 Define

Define is the TDMS operation through which the user names and describes the structural relationships of his data. It is the first operation the user must perform when he actually starts to use TDMS.

Virtually all of the capabilities of Define that were scheduled for implementation were working reasonably well at the beginning of this reporting period. Most of the effort in this operation has been on correcting errors that were uncovered as users tried out all the flexibility that was included in Define.

3.2.2 Generate

Generate (which was formerly known as Load) accepts all types of data, in either the batch or interactive mode, performs necessary validity checks, and allows on-line error correction, if desired. The output of the Generate operation is a TDMS data base, a self-defined file with self-defined entries and a completely inverted cross-index.

At the beginning of the reporting period, Generate was just beginning to be run successfully on medium-sized data bases. By the end of the reporting period, a number of large (over 1 million-byte) data bases had been generated, and--for standard operations--users were able to handle Generate with relative ease and a minimum of assistance.

A number of significant features were incorporated in Generate during this reporting period, including VALUE and FORMAT checking, on-line typewriter input, on-line error correction, and the ability to suspend and restart. Considerable effort was also expended on developing the ADDON operation, which was planned for inclusion in Version 5. This capability is operational, but is not checked out well enough to release. It is expected to be shipped with Version 5.1 early in the next reporting period.

Although Generate is capable of producing TDMS data bases, it currently places some limitations on users who want to build large data bases within narrow time constraints. This situation is receiving a good deal of attention and will be alleviated in the near future.

3.2.3 Query

Query allows TDMS users to retrieve data from a TDMS data base in an ad hoc fashion. The major commands--SHOW, PRINT, and DESCRIBE--were working well before the release of Version 3. This operation has been quite extensively used, and the number of reported errors has been very small; these errors have been fixed quickly. During this reporting period, the main Query capabilities were included in the Compose, Update, Maintain, and Display operations, and the following new features were incorporated:

- PRINT commands are now available without a WHERE or FOR clause.
- FORMAT and VALUE output is now included in DESCRIBE.
- Conditional expressions may now contain multiple HAS modifiers.
- The SHOW logic has been improved and actual values may be used in addition to V-numbers.
- Internal code and logic improvements have been made to speed up all phases of the operation.

In connection with the work on Query, a procedure called GETDATA has been developed which permits JOVIAL programmers to easily retrieve data from a TDMS data base for special-purpose manipulation. This procedure was released on schedule with Version 5.

3.2.4 Compose

Compose is a comprehensive report generator. It is a two-phase operation: the first phase is that of designing a report, in which Compose acts in a highly interactive fashion with the user to design a report format according to user instructions. The language used in doing this job is extremely powerful, and yet has proved to be relatively easy for people to learn. The second phase is the actual production of the desired report; it is analogous to a compile-and-go operation.

Although most of Compose was operational for the Version 3 release of TDMS, several important features were not available until the release of Version 5. The task of implementing these features was emphasized during this reporting period. Among the new features included through Version 5 are:

- A DLO capability
- Sensitivity to page width of the interactive console

- Multiple SORT variables
- Implementation of the format commands: SPACE, FEED, SHIFT, and MASK
- A report modification capability
- Implementation of the LINK command
- Implementation of derived variables

In addition to implementing these features, a considerable increase in the running speed of Compose was attained with the release of Version 5.

During the coming year, a number of features will be added to the Compose operation, including:

- The LIMIT command
- The PUT command
- The 'K token and accompanying features
- The ability to use elements as operands in conditional expressions anywhere except in a QUALIFY command
- The PER DISTINCT modifier in statistical expressions
- The TABLE command
- The use of modified statistical expressions within derived variables

3.2.5 Update

Update permits a user to dynamically change, add, or delete single values or entire entries in a data base as it exists on the disc. This operation is an extremely complex job, requiring linkage changes throughout the data base every time an Update command is executed.

Most of Update was working in Version 3, with the exception of the capabilities to add and remove repeating groups, and to check formats and values on new data items. These limitations have been corrected, and the work on Update is now mainly one of tracking down discrepancy reports and correcting errors brought about by unforeseen conditions.

3.2.6 Maintain

Maintain represents a significant research effort within TDMS. The intention of this operation is to provide the user with a generalized means for merging, subsetting, extracting, ordering, restructuring, and updating data bases. The Maintain programs are designed to accept one or two different data bases, and an on-line description of the desired output data base, rules for the selection of data, and the transformations that are required. As with Compose, the capability to interactively describe a task on-line, save it for repetitive use, and modify it as desired has been included.

Of the eight tasks originally isolated as a meaningful set of generalized maintenance tasks, only Batch Update and Copy-and-Cleanup have been scheduled for inclusion in initial versions of TDMS. The Copy-and-Cleanup task is required to reallocate spares in a data base that has been extensively modified by use of the Update operation. This task completed checkout in January and was released with Version 5.

The Batch Update task is aimed at handling large volumes of changes rather than the small volumes expected to be handled on-line by Update. Most of the effort during this reporting period has been aimed at getting the Batch Update task checked out and working on available test cases. The condition of Batch Update is now such that users may begin to use it successfully; in some cases it is anticipated that users will uncover errors which a lack of sufficient test materials precludes uncovering during checkout. This operation will undoubtedly require a good deal of user experience to help "fine tune" it. All of the capabilities that are necessary for full operation have been coded.

At the end of this reporting period, work had begun on the design of a third task (Subset).

3.3.7 Reformat

Reformat is intended to provide a straightforward method of converting symbolic data that already exists in machine-readable form into the numbered field data set format required by the TDMS Generate operation. The basic capabilities for handling both standard fixed fields and indentured fixed-field input types have been checked out and are working in both the AMCSSC and the Air Force Command test. This capability was achieved with the aid of several trips to the Washington area. The trips were used mainly to help operational users with procedural and program problems. Due to the availability of other tools (like DbL) that accomplish the same purpose as Reformat, it is unclear whether further effort on Reformat as it exists will be required.

3.3.8 Data Base Oriented Programming Language

This work was aimed at providing a bridge between the computational capabilities of JOVIAL and the data base manipulation ability of TDMS. A procedure called GETDATA was developed; it permits JOVIAL programmers to retrieve data from a TDMS data base for special-purpose manipulation. This procedure can be compiled as part of a JOVIAL program. Since this procedure utilizes the Query language of TDMS, users need not learn new language forms or expressions. GETDATA was released on schedule with Version 5 of TDMS.

3.2.9 META5 Adaptation

A special-purpose language (known as DBL) has been adapted from META5* for the purpose of reformatting symbolic data bases into a form acceptable to TDMS. DBL is intended for non-programmers, and thus is relatively non-procedural in its use. It allows the user to describe the format of a data base and those format conversions he wishes to effect. He need not be well versed in TDMS conventions, since the burden of actually performing the conversions is borne by the DBL compiler. The language is simple enough to be learned in a few hours for most purposes.

Currently, compilers for DBL exist on the IBM 360/50 and 4-38 computers, and produce output suitable for TSS-LUCID and TDMS, the output form selected depends on the compiler selected. Numerous data bases were successfully converted during this reporting period by users at the NMCSSC, Air Force Command Post, Air Force Academy, and the Southwest Regional Educational Laboratory.

The DBL language provides the user with all of the linguistic features originally intended. These features include:

- An ALGOL-like IF-statement with the full complement of logical and relational operators
- Infix integer arithmetic
- Loop and iteration control (with Boolean expressions where desired)
- Conversion lists so that data may be coded or decoded
- Implicit and explicit output capabilities
- Data structures such as simple variables and pushdown stacks
- Full META5 capability to perform any tasks not possible in DBL, but which could normally be coded in META5
- Programs compiled in one pass directly to META5 assembly language

Major emphasis during this reporting period was placed on checkout of the various DBL compilers. Operational experience gained during this period indicated the need for more detailed documentation on DBL and META5. This work is being accomplished at present. Since the implementation of the one-pass compiler, the program's overall efficiency was markedly improved--indeed, it is often possible to produce more efficient code for certain kinds of character scanning in DBL than in META5.

At present, the META5 capability in DBL is adequate to process any free format sections in data base reformatting. The generation of fixed-format output appears to be quite difficult with META5 as it presently stands, and so that capability has not been added to DBL.

---

*META5 was renamed to META5 during a previous reporting period.

Improved error message and trace capabilities were coded for META6/360 to aid in debugging. Error messages now give the name of the rule in which the error occurred, in addition to the information previously given. The trace capability indicates to the user when a rule has been entered, and the truth value of the rule upon exiting it. These same META6 debugging aids can also be utilized by those DBL users who are familiar with META6 and its conventions.

3.3 STAFF

A. H. Vorhaus, Staff Head
Beverly Bourne, Technical Assistant
Carla Mapes

TDMS User Documentation Project

R. H. Barsalou, Project Leader
W. D. Dinges
D. J. Mecham
Ann Reisman
J. M. Sheldon

TDMS Language Project

J. D. Reynolds, Project Leader
Nora Bosak
D. P. Crowson
D. D. Fess
D. W. Glenn
Celeste Hand
H. W. Henley
B. P. Landell
Alice Lanes
J. F. Murray
Ellen Ossofsky
Jean Saylor
Jean Schiess
S. Shapiro
W. A. Shonborn
R. P. Stewart

TDMS Preparation & Maintenance Project

C. A. Kribs, Project Leader
K. M. Brandon
Anne Cady
R. C. Catlett
T. J. Champion
P. A. DeSimone
J. Farell
Leah Fine
R. Gates
Ann Kligman
M. Peltz
R. B. Reynolds
Margaret Rybak
W. A. Shasberger
W. H. Stone

META5 Adaptation

M. Schaefer

3.4 DOCUMENTATION

This section lists those documents describing the data management portion of the ADEPT system that were produced during this reporting period. Some of the documents listed below were released in SDC's "Note" series. These documents are internal working papers only, and have not been cleared for open publication.

Rybak, M. A. Implementation of MORE *system:term*. SDC document N-(L)-23550/101/00. 30 September 1968. 6 pp.

Describes a new procedure, MORST, which has been added to SCAN for implementing the feature MORE *system:term* in TDMS programs.

Peltz, M. TDMS data base checker. SDC document N-(L)-23550/180/00. 29 August 1968. 5 pp.

Describes the TDMS Data Base Checker package, which is designed to offer a user a variety of procedures, at his own discretion, to check the validity of his data base and also to provide the validity of his data base and also to provide for an on-line printout of errors discovered and related entry data.

Peltz, M. TDMS data base printer. SDC document N-(L)-23550/181/00. 29 August 1968. 5 pp.

Describes the TDMS Data Base Printer package, which is designed to provide an off-line or on-line printout of any or all of the central tables of a TDMS data base. At the user's discretion, selected records from any one central table can also be selected for printing.

Peltz, M. and Rybak, M. A. EMOV: multiple disc file to tape copy program. SDC document N-(L)-23550/607/01. 7 October 1968. 7 pp.

Describes EMOV, which provides a capability to copy onto a single tape reel up to 20 disc files under the ADEPT system. This program bypasses the single file per tape limitation currently in ADEPT by organizing the disc files as subfiles (groups of records) on the tape.

DeSimone, P. A. The language specifications for the define operation of TDMS. SDC document TM-3370/003/01. 7 October 1968. 23 pp.

Describes the Define operation of TDMS, which is used to describe new data bases and to modify existing data base descriptions. The language of the Define operation is specified, and the user's interaction with the operation is described.

Barsalou, R. H. The time-shared data management system interim user's guides. SDC document TM-3849/000/02. 26 August 1968. 3 pp.

Establishes a documentation series for TDMS. This series includes a set of interim user's guides for all TDMS operations.

Barsalou, R. H. DEFINE interim user's guide. SDC document TM-3849/002/02. 26 August 1968. 19 pp.

Provides instructions for propsective users of the DEFINE operation during the period of initial familiarization.

Barsalou, R. H. SIZES interim user's guide. SDC document TM-3849/008/00. 23 August 1968. 8 pp.

Provides instructions for propsective users of the SIZES program during the period of initial familiarization.

Fess, D. D. RETDMS interface specifications. SDC document TM-4131. 18 November 1968. 42 pp.

Instructs the prospective user on how to use RETDMS. The purpose of RETDMS is to process user input statements, identify TDMS language forms within the input statement string, and provide identifying output component values and information using data selected from the input TDMS data base.

Bleier, R. E. and Vorhaus, A. H. File organization in the SDC time-shared data management system (TDMS). SDC document SP-2907. 1 August 1968. 17 pp.

Describes the reason for the development of generalized file-processing systems and shows how file organization affects the overall functioning of file-processing systems. The Time-Shared Data Management System (TDMS) is used to illustrate one of these file organizations.

4. PROGRAMMER'S PACKAGE

4.1 INTRODUCTION

The programmer's package developed for use in the ADEPT system contains four elements for the professional programmer: a compiler, a debugging capability, an editing capability and a utility program. A teletype interpreter, known as TINT, is also provided for the user who is not a professional programmer. In addition, an Interactive Programming Support System--which combines these capabilities into a single system--was developed.

4.1.1 JOVIAL Compiler

JOVIAL is a general-purpose programming language well suited for a variety of different applications, including scientific and engineering problems involving numeric computation, administrative problems involving large data files, and logically complex problems involving symbolic data. Because of the optional control it provides over the details of storage allocation, JOVIAL is especially suitable for problems requiring an optimum balance between data storage and program execution time.

The JOVIAL compiler developed for use by the professional programmer in the ADEPT system has all of the capabilities of Basic JOVIAL, as well as several extra features requested by the initial users of the system. Some of these additional capabilities include provision for longer literals and the ability to compile program segments independently and then combine these segments at execution time. This compiler is compatible with a JOVIAL compiler delivered to the initial system users, which runs under their (OS/360) operating system.

4.1.2 Debugging Aid

The debugging program developed as part of ADEPT provides an on-line capability for the professional programmer to look at and change his program and program data during execution, and to switch between "execution" and "look-and-change" modes.

4.1.3 Editing Aid

The editing program provided in ADEPT gives the professional programmer an on-line capability to maintain and modify his program in source language. It may also be used to generate original code. The editing commands available at present are: COMPOSE, INSERT, REPLACE, DELETE, MOVE, COPY, SEQUENCE, and CHANGE. Also available are the utility commands: DISPLAY, SAVE, and QUIT.

4.1.4 Utility Program

The utility program provides some basic program maintenance services needed by the programmer--namely, card-to-tape and tape-to-printer and punch conversions.

4.1.5 TINT (Teletype Interpreter)

TINT is an interactive programming system aimed at the casual programmer who has small-to-moderate sized programs. It uses a dialect of JOVIAL, and combines techniques of compilation and interpretation.

TINT is designed to bridge the language gap between the computer and the non-programmer user; it operates interpretively, on-line. The on-line nature of the system makes it convenient to use. With TINT, the user can create, check out, execute, modify and re-execute a program directly from a remote teletype--and can often carry out the entire programming process at a single sitting. TINT is particularly suited to compact programs such as short mathematical problems, subprogram checkout, and other "one-shot" operations.

4.1.6 Interactive Programming Support System (IPSS)

The goal of the Interactive Programming Support System (IPSS) is to permit all of the programming processes--composition, editing, execution, testing and documentation--to be carried out as parts of a single, coordinated activity centered around an interactive compiler. The system unifies techniques that are usually embodied in separate functional programs so that the programmer need not know which particular program is performing a specific task. The system, of course, is intended for a time-sharing environment, with user interaction via a CRT/keyboard console or teletypewriter.

4.2 PROGRESS

Since all of the components of the ADEPT-50 programmer's package were operational prior to this reporting period, most of the effort on the programmer's package during this period was devoted to shaking down programs, removing program errors, improving functional capabilities, and documenting program usage. The programmer's package was installed at several new military facilities during this reporting period, as part of the total ADEPT-50 system. Thus far, use of this portion of the system has been low, since most ADEPT-50 installations have been concentrating on learning to operate TDMS and the ADEPT-50 executive. The programmer's package was exclusively used during this period, however, by members of the ADEPT development team in Santa Monica. Relatively few program "bugs" were found, and these were repaired quickly. Several discrepancies in documents were reported; these have been corrected. A number of users have suggested ways of improving the usefulness of programmer's package components; these suggestions are being evaluated.

Some effort was also devoted to creating a standard format for identifying files, and a standard set of default options. Most of the components of the programmer's package have been modified (or are being modified) to use these standard forms.

Finally, a number of utility programs and JOVIAL library procedures were developed in this reporting period, most of which go beyond the scope of the original contract. These programs are typically generated by ADEPT application programmers as an open-ended set. Due to resource limitations, some of these procedures will have to be made available to the ADEPT community without full maintenance support.

4.2.1 JOVIAL Compiler

Shakedown of the JOVIAL compiler by ADEPT users turned up a number of errors which were corrected. The compiler staff also spent some time working with users to find their errors of usage, as well as compiler-induced errors. Cleanup of the various interfaces between the compiler and the executive, and among the segments of the compiler itself received considerable attention. Several improvements in the quality of the object code were made, and a substantial improvement in code which branches forward was undertaken. Some improvement in performance was obtained by increasing the tape input/output buffer size.

The standard library for use during compilation was augmented by a large number of procedures which perform simple tasks for the programmer. One series of routines covers a number of aspects of input/output and makes it easier for the programmer to deal with the executive's Cataloger and SPAM programs. Included among these routines are:

- Read or Write Interactive Terminal
- Scan EBCDIC Image
- Standard (I/O) Default Procedure
- Set Cataloger Table File Identification Values
- Check SPAM Return Codes
- SPAM Call
- Cataloger Call
- Check Results of Cataloger Operation
- Position Opened File
- Close File
- Open File
- Get Cataloger Table and Open or Close File
- Read Next Sequential File Record
- Write Next Sequential File Record

A second set of procedures provides string manipulation capability. Among these procedures are:

- Move Byte String
- Compare Byte Strings
- Exchange Byte Strings
- Set Byte String
- Zero Byte String

4.2.2 Debugging Aid

The debugging aid has been almost fault-free. It has received considerable use and only a few very minor bugs turned up. The debugging function has been split in two parts, to meet space requirements: one part resides in core as part of the Basic Executive, while the larger part is stored with the Extended Executive.

4.2.3 Editing Aid

The editing aid has received minimal shakedown usage, mostly by the NMCSSC and ADEPT installation team. A few problem areas have turned up and are being resolved. All of these were considered low priority problems by the users.

4.2.4 Utility Program

The basic utility program received very little usage. Programmers seemed to prefer other programs which had fewer features, but were easier to use. A number of bugs were found in these auxiliary programs, and since they were not written as a set (like the basic utility program itself), several incompatibilities were found. Consequently, the utility area was reexamined, and considerable work was put into developing and improving this area. DEBE--a program which moves data between the card reader and other storage media--was considerably reworked. LISTER and FDMP (File Dump)--programs for printing various files--were written and debugged. A program for producing a programmer-specific library of files on tape was also written and checked out. Finally, a tape-to-tape COPY program was developed, and is now being extended.

4.2.5 TINT

TINT has been modified so that saved programs can be changed with the editing aid. This involved moving the location of the sequence number field. Work was begun on making TINT more useful to JOVIAL programmers by providing a command that will produce a copy of a program written in TINT which is suitable for compilation. Most of the incompatibilities have been taken care of, and work is presently going into the READ and WRITE areas.

4.2.6 Interactive Programming Support System (IPSS)

The program composition and editing phase of IPSS was used by the staff in Santa Monica and given some shakedown by the installation team in Washington. A number of errors were turned up and fixed. The IPSS program itself was run through IPSS.

Some work was invested in shortening various parts of the program to improve performance or meet space requirements. Several new capabilities were added to IPSS. These include:

Automatic output of set/use information when the set/used table is filled, so that the table can be reused for new information.

- Provision for accepting statements continued in a separate input either off-line or on-line.
- Improved error detection in the edit mode. (Sequences of edited statements are treated as in the compose mode, and a memory of expected sequences is maintained.)
- The possibility of changing a procedure declaration (number of parameters or from procedure to function) was added. Such changes were formerly impossible in IPSS.
- Acceptance of ditto marks in input statements to reduce repetitious typing.
- Provision for hard copies of the CRT either on-line or off-line.

Work was begun on the next phase of IPSS--hooking compilation and partial recompilation capability to that of program composition and editing. A one-pass compiler under development by other members of the programmer's package staff was studied for its applicability to IPSS. Also under study was the possibility of using the translator phase of the regular JOVIAL compiler and modifying the existing IPSS routines to perform the work normally done by a compiler generator. The main problems are in providing the translator with a partial recompilation capability.

4.3 STAFF

E. H. Jacobs, Staff Head
- H. Bratman, Compiler Systems Project Leader
  - G. M. Cady
  - H. G. Martin
  - R. N. Mathur
  - Ellen C. Perstein
  - D. W. Ranum
  - N. A. Sandin

4.4 DOCUMENTATION

This section lists those documents describing the programmer's package portion of the ADEPT system that were produced during this reporting period. Some of the documents listed below were released in SDC's "Note" series. These documents are internal working papers only, and have not been cleared for open publication.

Sandin, N. A. JOVIAL (prJ5.3) compiler user's guide. SDC document N-23729/001/00. 5 November 1968. 84 pp.

Describes the use of the ADEPT-50 JOVIAL compiler. Includes a general description of the compiler, a brief discussion of the language forms processed, specific information for inputs, descriptions and samples of outputs, file structures and locations.

McCabe, J. D. Move byte string procedure, STRINGM. SDC document N-23729/802/01. 15 November 1968. 1 p.

Describes the use of JOVIAL procedure STRINGM.

Martin, H. G. Coded hollerith to integer function, NTGR. SDC document N-23729/803/00. 15 July 1968. 2 pp.

Describes the use of JOVIAL procedure NTGR.

Martin, H. G. Read or write interactive terminal, TRMNL. SDC document N-23729/804/00. 15 July 1968. 2 pp.

Describes the use of JOVIAL procedure TRMNL.

Martin, H. G. Scan EBCDIC image, SKAN. SDC document N-23729/805/00. 15 July 1968. 2 pp.

Describes the use of JOVIAL procedure SKAN.

Mathur, R. N., A standard default procedure, DEFAULT. SDC document N-23729/807/00. 13 August 1968. 2 pp.

Describes the use of JOVIAL procedure DEFAULT.

Martin, H. G. Set cataloger table file identification values. SDC document N-23729/808/00. 25 September 1968. 3 pp.

Describes the use of JOVIAL procedure SETFIV.

Martin, H. G. Check SPAM return codes, SPAMCK. SDC document N-23729/809/00. 23 October 1968. 2 pp.

Describes the use of JOVIAL procedure SPAMCK.

Martin, H. G. SPAM call. SDC document N-23729/810/00. 23 October 1968. 2 pp

Describes the use of JOVIAL procedure SPAMC.

Martin, H. G. Cataloger call, KAT. SDC document N-23729/811/00. 23 October 1968. 2 pp.

Describes the use of JOVIAL procedure KAT.

Martin, H. G. Check results of cataloger operation. SDC document N-23729/812/00. 23 October 1968. 2 pp.

Describes the use of JOVIAL procedure CATCHK.

Martin, H. G. Position opened file. SDC document N-23729/813/00. 23 October 1968. 2 pp.

Describes the use of JOVIAL procedure POSISHUN.

McCabe, J. D. Compare byte strings procedure, STRINGC. SDC document N-23729/814/00. 1 November 1968. 2 pp.

Describes the use of JOVIAL procedure STRINGC.

McCabe, J. D. Exchange byte strings procedure, STRINGX. SDC document N-23729/815/00. 15 November 1968. 1 p.

Describes the use of JOVIAL procedure STRINGX.

McCabe, J. D. Set byte string procedure, STRINGS. SDC document N-23729/816/00. 15 November 1968. 1 p.

Describes the use of JOVIAL procedure STRINGS.

McCabe, J. D. Zero byte string procedure, STRINGZ. SDC document N-23729/817/00. 15 November 1968. 1 p.

Describes the use of JOVIAL procedure STRINGZ.

Martin, H. G. Close a file, CLOZ. SDC document N-23729/818/00. 26 November 1968. 2 pp.

Describes the use of JOVIAL procedure CLOZ.

Cady, G. M. Cataloger call procedure, ADPCAT. SDC document N-23729/819/00. 11 December 1968. 1 p.

Describes the use of JOVIAL procedure ADPCAT.

Martin, H. G. Get cataloger table and open or close file, GETRQT. SDC document N-23729/820/00. 9 January 1969. 4 pp.

Describes the use of JOVIAL procedure GETRQT.

Martin, H. G. Open file, OPENF. SDC document N-23729/821/00. 14 January 1969. 2 pp.

Describes the use of JOVIAL procedure OPENF.

Martin, H. G. Read next sequential file record, READS. SDC document N-23729/822/00. 10 January 1969. 4 pp.

Describes the use of JOVIAL procedure READS.

Martin, H. G. Write next record in sequential file, WRITS. SDC document N-23729/823/00. 10 January 1969. 3 pp.

Describes the use of JOVIAL procedure WRITS.

Bratman, H. and Perstein, E. C. User's guide for IPSS--Interactive programming support system. SDC document N-23759/200/03. 7 October 1968. 31 pp.

Describes the use of IPSS to compose, syntax-check, and edit JOVIAL programs. Includes a description of applicable IPSS commands.

Perstein, E. C. and Bratman, H. User's guide for IPSS (interactive programming support system). Using the IBM 2250 CRT display. SDC document N-23759/201/00. 30 September 1968. 13 pp.

Describes the use of IPSS on the IBM 2250 CRT display. Includes a description of applicable commands.

Swerdlow, S. G. User's guide for DEBE (ADEPT card file handling utility program). SDC document N-23759/300/02. 25 October 1968. 5 pp.

Describes the use of DEBE for copying card files from the card reader, tape, or disc storage onto the printer, card punch, tape, or disc.

Swerdlow, S. G. LISTER user's guide. SDC document N-23759/301/01. 15 November 1968. 4 pp.

Describes the use of LISTER, an ADEPT-50 utility program that permits the user to list DLO files under time-sharing. Includes a discussion of acceptable file structure and options.

Martin, H. G. User's guide for FDMP, file dump utility program. SDC document N-23759/310/01. 10 February 1969. 15 pp.

Describes the use of FDMP for listing tape or disc file records on-line or off-line in EBCDIC or hexadecimal form.

Martin, H. G. GULF (generate, update, list and fetch from a library file) user's guide. SDC document N-23759/330/00. 14 January 1969. 10 pp.

Describes the use of GULF for generating, updating, listing and fetching sequential files to or from a library of files.

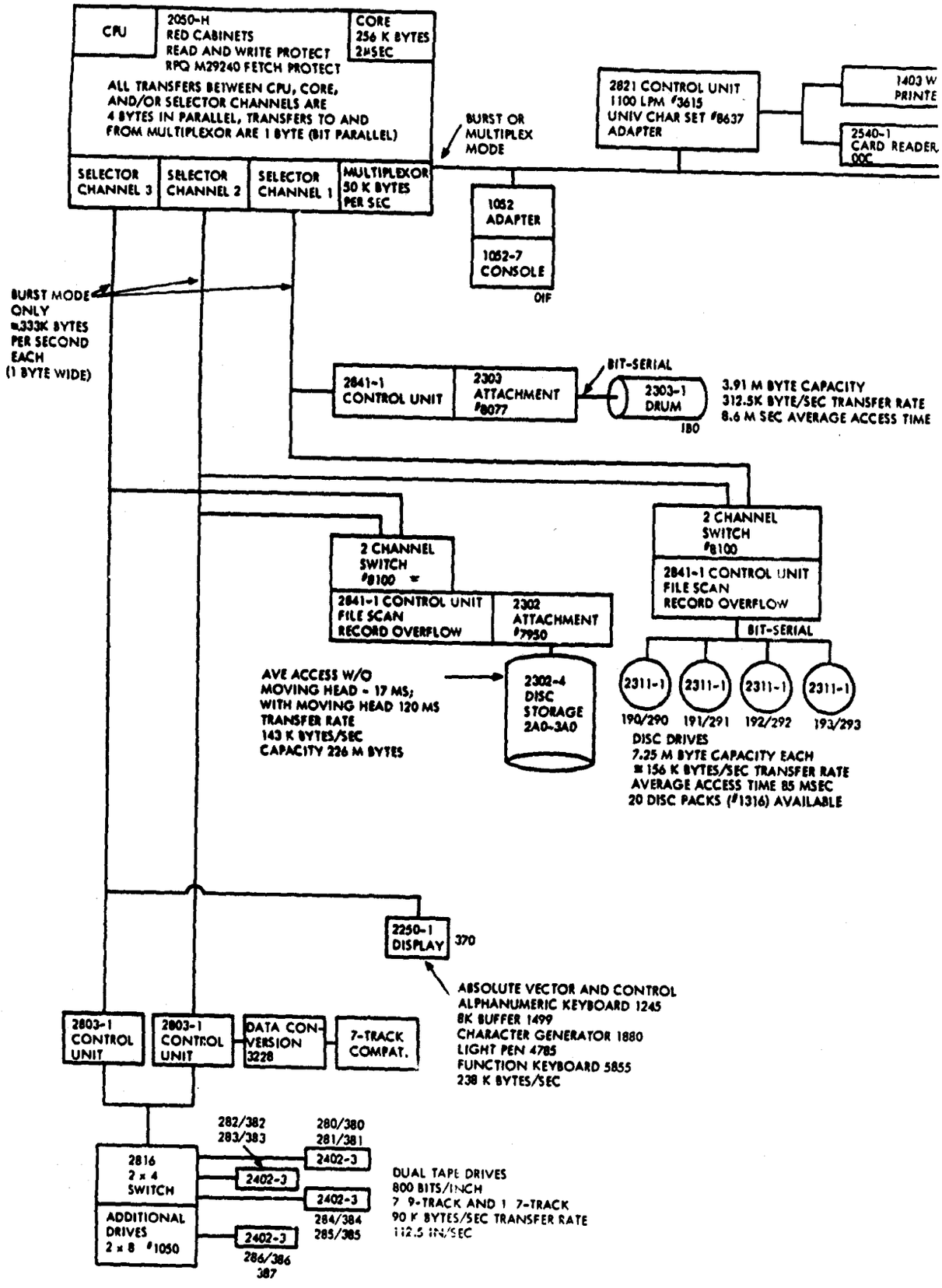
Martin, H. G. File copy program (COPY) user's guide. SDC document N-23759/340/00. 1 August 1968. 23 pp.

Describes the use of the file copy program, COPY, for copying files fully or partially to or from any available storage volume (except on-line printers or card devices).

Kennedy, P. K. The TINT1.1 user's guide. SDC document TM-3881-502/00. 1 August 1968. 179 pp.

A self-instructional guide to TINT1.1, an interactive programming system operating under control of the ADEPT-50 executive.

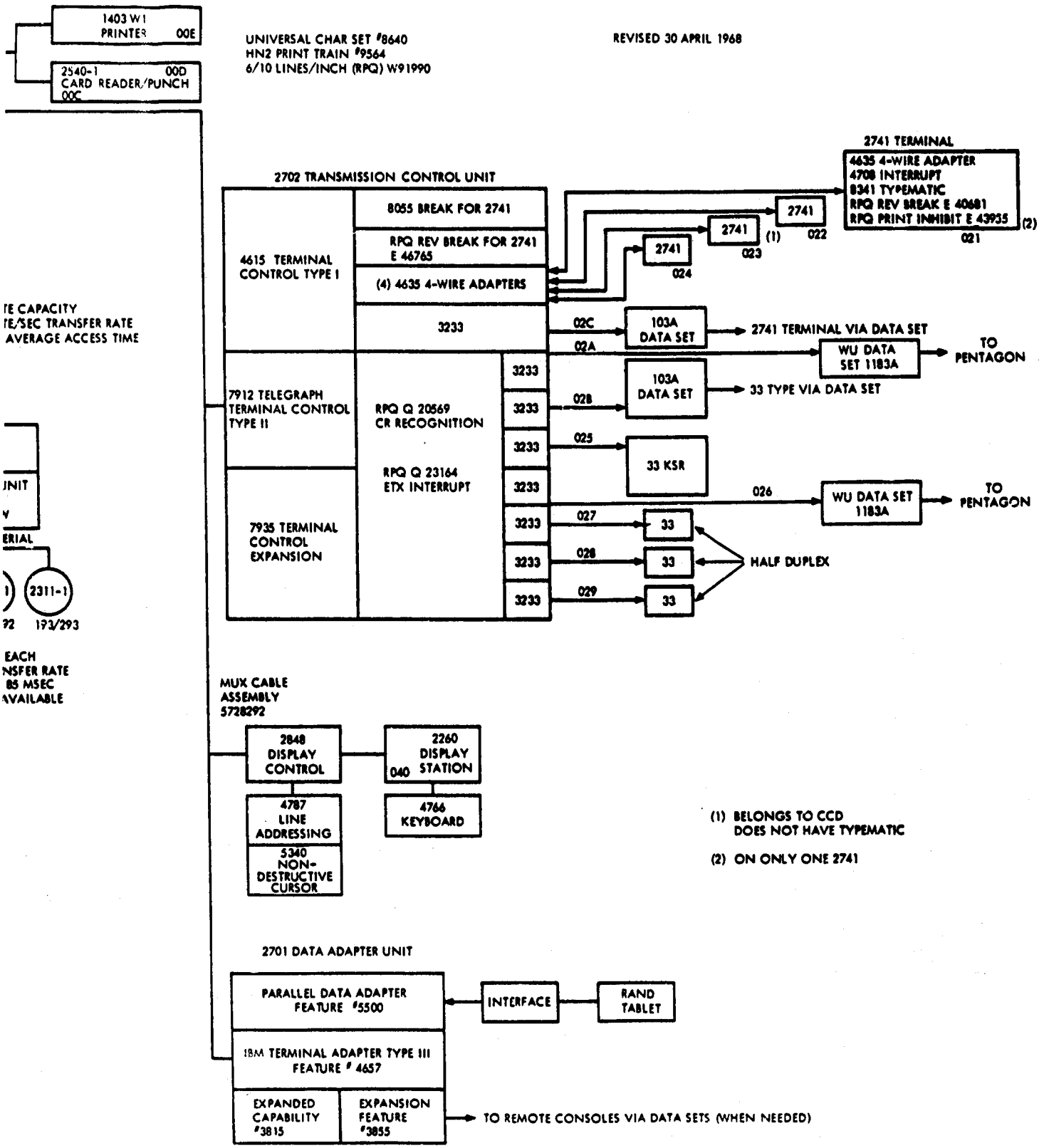
Kratman, H., Martin, H. G., and Perstein, E. D. Program compostion and editing with an on-line display. Proceedings, 1969 Fall Joint Computer Conference, December 1968. pp. 1349-1360.


CPU
2050-H
RED CABINETS
READ AND WRITE PROTECT
RPQ M29240 FETCH PROTECT
CORE
256 K BYTES
2μSEC
ALL TRANSFERS BETWEEN CPU, CORE, AND/OR SELECTOR CHANNELS ARE 4 BYTES IN PARALLEL, TRANSFERS TO AND FROM MULTIPLEXOR ARE 1 BYTE (BIT PARALLEL)
BURST OR MULTIPLEX MODE
2821 CONTROL UNIT
1100 LPM #3615
UNIV CHAR SET #8637
ADAPTER
1403 W
PRINTE
2540-1
CARD READER
00C
SELECTOR CHANNEL 3
SELECTOR CHANNEL 2
SELECTOR CHANNEL 1
MULTIPLEXOR 50 K BYTES PER SEC
1052 ADAPTER
1052-7 CONSOLE
01F
BURST MODE ONLY
≈333K BYTES PER SECOND EACH
(1 BYTE WIDE)
2841-1 CONTROL UNIT
2303 ATTACHMENT #8077
BIT-SERIAL
2303-1 DRUM
180
3.91 M BYTE CAPACITY
312.5K BYTE/SEC TRANSFER RATE
8.6 M SEC AVERAGE ACCESS TIME
2 CHANNEL SWITCH #8100
2841-1 CONTROL UNIT FILE SCAN RECORD OVERFLOW
2302 ATTACHMENT #7950
2 CHANNEL SWITCH #8100
2841-1 CONTROL UNIT FILE SCAN RECORD OVERFLOW
BIT-SERIAL
AVE ACCESS W/O MOVING HEAD - 17 MS; WITH MOVING HEAD 120 MS
TRANSFER RATE 143 K BYTES/SEC
CAPACITY 226 M BYTES
2302-4 DISC STORAGE 2A0-3A0
2311-1
2311-1
2311-1
2311-1
190/290
191/291
192/292
193/293
DISC DRIVES
7.25 M BYTE CAPACITY EACH
≈ 156 K BYTES/SEC TRANSFER RATE
AVERAGE ACCESS TIME 85 MSEC
20 DISC PACKS (#1316) AVAILABLE
2250-1 DISPLAY
370
ABSOLUTE VECTOR AND CONTROL
ALPHANUMERIC KEYBOARD 1245
8K BUFFER 1499
CHARACTER GENERATOR 1880
LIGHT PEN 4785
FUNCTION KEYBOARD 5855
238 K BYTES/SEC
2803-1 CONTROL UNIT
2803-1 CONTROL UNIT
DATA CONVERSION 3228
7-TRACK COMPAT.
2816 2 x 4 SWITCH
ADDITIONAL DRIVES 2 x 8 #1050
282/382
283/383
280/380
281/381
2402-3
2402-3
2402-3
2402-3
284/384
285/385
286/386
387
DUAL TAPE DRIVES
800 BITS/INCH
7 9-TRACK AND 1 7-TRACK
90 K BYTES/SEC TRANSFER RATE
112.5 IN/SEC

A

DEVELOPMENT PROTOTYPE SYSTEM BLOCK DIAGRAM

TM-3628/003/00



B

DOCUMENT CONTROL DATA - R&D

System Development Corporation
Santa Monica, California

2a. REPORT SECURITY CLASSIFICATION
Unclassified
2b. GROUP

3. REPORT TITLE

Advanced Development Prototype Final Report (Part 1) to the Director, Advanced Research Projects Agency for the Period 30 July 1968 to 30 January 1969.

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

5. AUTHOR(S) (First name, middle initial, last name)

J. I. Schwartz, Project Director

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
30 January 1969	56	

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
DAHC15-67-C-0149	TM-3628/003/00
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. DISTRIBUTION STATEMENT

Distribution of this document is unlimited

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY

13. ABSTRACT

This final report describes work done under Part 1 of the Advanced Development Prototype contract from 30 July 1968 to 30 January 1969. The result of this work is ADEPT--a comprehensive information-processing system implemented at SDC for operation on IBM 360 computers. This report includes an overview of the current status of the system, and a detailed description of the three major components of ADEPT; a time-sharing executive, a data management component (consisting mainly of the Time-Shared Data Management System), and a programmer's package, which includes a JOVIAL compiler, editing, debugging, and utility programs, a teletype interpreter (TINT), and an Interactive Programming Support System. Also included in this document are the names of staff members assigned to each of the three major project areas, as well as a listing of the documents produced in each area during this reporting period. Upon request, referenced documents will be made available to appropriate organizations.

DD FORM 1 NOV 65 1473

Unclassified
Security Classification

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Advanced Development Projects (ADP) ADPT Information processing Time-sharing Data management JOVIAL compiler Teletype interpreter (TINT) TINT IBM 360 Interactive Programming Support System Time Data Management System TDMS						